Sparse Kernel Machines

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

March 12, 2019

SVM - Contents

- SVM for Regression, SVM for Classification, Linear Classifiers, Classifier Margin, Support Vectors, Computing the margin width
- SVM for the non-separable case
- Probabilistic Interpretation of SVM, SVM Vs. Logistic Regression
- SVM for the Separable Case, Kernel Trick and Non-Linear Classifier, Regularization and Kernel Parameter Estimation, Probabilistic Response, Multiclass classification

Following closely:
Bishop CM, *Pattern Recognition and Machine Learning*, Springer, 2006 (Chapter 7)
Murphy, K. Machine Learning: A probabilistic Approach (Chapter 14)

RVM - Contents

- Relevance Vector Machines: Likelihood, Prior, Posterior, Predictive Distribution
- Evidence Approximation, Evidence, Two useful matrix identities
- Fast training algorithm
- Splitting the evidence, Analysis of the evidence
- Complete algorithm, Example
- Fast updates
- Extenstion to Multiple Outputs, Numerical Stability Concerns

Support Vector Machines

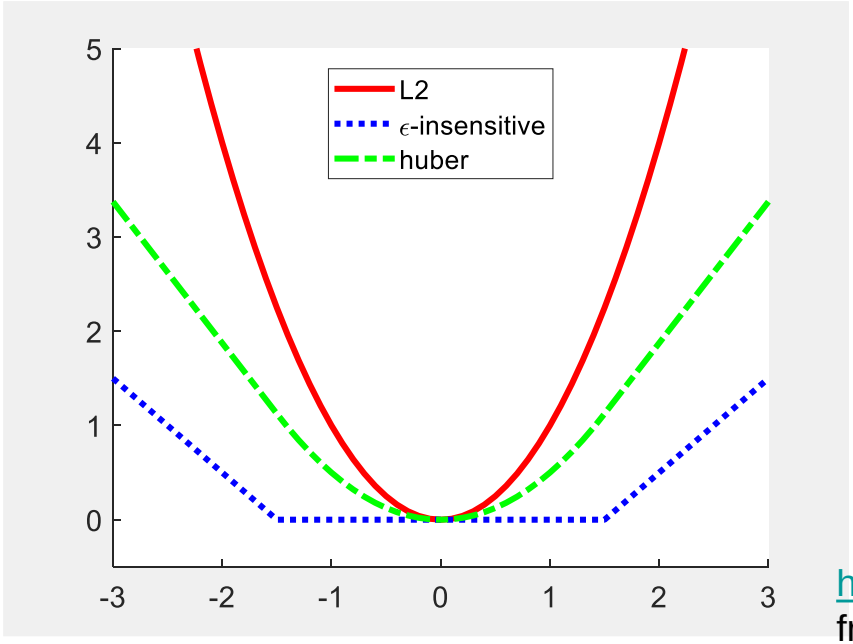
SVMs for Regression

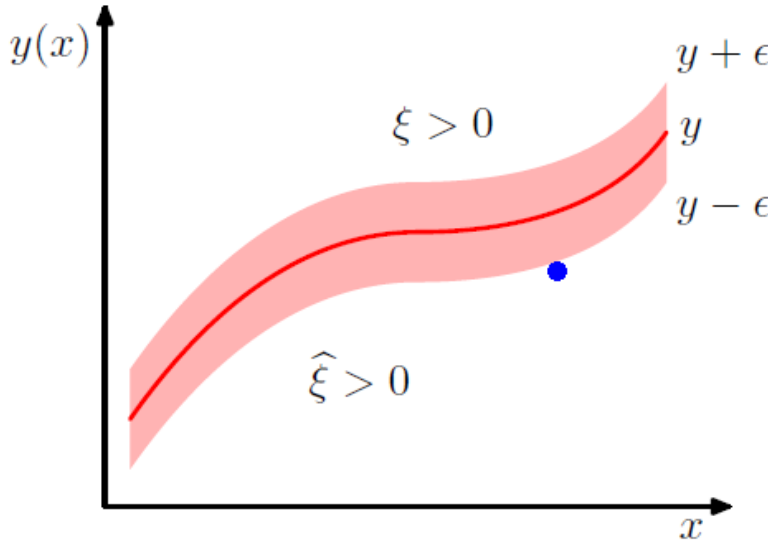
- In our approach to regression, $\boldsymbol{w}$ depends on all training inputs. We are here looking for sparse approximations.

- Consider the ϵ –insensitive loss function: $L_\varepsilon(y,\hat{y}) \equiv \begin{cases} 0 & if \quad |y-\hat{y}| < \varepsilon \\ |y-\hat{y}| - \epsilon & if \quad |y-\hat{y}| > \varepsilon \end{cases}$

  leading to the following objective function:

$$\mathrm{J} = \mathrm{C}\sum_{i=1}^{N} L_\varepsilon(y_i,\hat{y}_i) + \frac{1}{2}\|\boldsymbol{w}\|^2,\ \hat{y}_i = f(\boldsymbol{x}_i) = w^T\boldsymbol{x}_i + w_0, C = 1/\lambda$$





huberLossDemo.m
from PMTK3

SVMs for Regression

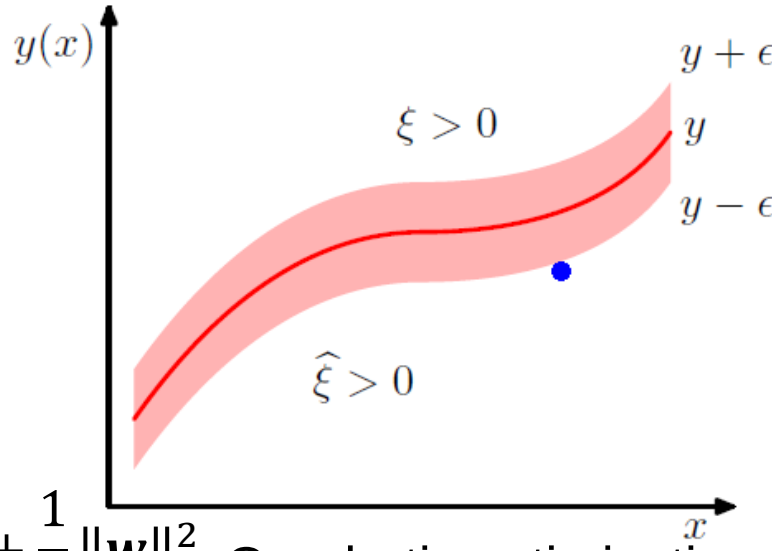
- In our approach to regression, w depends on all training inputs. We are here looking for sparse approximations.

- Consider the ϵ –insensitive loss function: $L_\varepsilon(y,\hat{y}) \equiv \begin{cases} 0 & if \ \ |y-\hat{y}| < \varepsilon \\ |y-\hat{y}| - \epsilon & if \ \ |y-\hat{y}| > \varepsilon \end{cases}$

  leading to the following objective function:

$$\mathrm{J} = \mathrm{C}\sum_{i=1}^{N} L_\varepsilon(y_i,\hat{y}_i) + \frac{1}{2}\|\boldsymbol{w}\|^2,\ \hat{y}_i = f(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i + w_0, C = 1/\lambda$$

- Introduce slack variables to describe the degree to which each point lies outside the ϵ – tube:



$$y_i \le f(\boldsymbol{x}_i) + \varepsilon + \xi_i^+, \xi_i^+ \ge 0, i = 1,\ldots,n$$
$$y_i \ge f(\boldsymbol{x}_i) - \varepsilon - \xi_i^-, \xi_i^- \ge 0, i = 1,\ldots,n$$

$$\min_{w,\xi_i^+\ge 0,\xi_i^-\ge 0} J = \min_{w,\xi_i^+\ge 0,\xi_i^-\ge 0} \mathrm{C}\sum_{i=1}^{N}(\xi_i^+ + \xi_i^-) + \frac{1}{2}\|\boldsymbol{w}\|^2$$

Quadratic optimization

$2N + D + 1$ variables

SVMs for Regression

- One can show that: $\widehat{\boldsymbol{w}} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i,\ \alpha_i \geq 0.$
- The $\boldsymbol{a}$ vector is sparse since we do not care about errors $\leq \varepsilon$. The $\boldsymbol{x}_i$'s are called support vectors – these are points for which the errors lie on or outside the ε tube.
- Making predictions:

$$\hat{y}(\boldsymbol{x}) = \widehat{w}_0 + \widehat{\boldsymbol{w}}^T \boldsymbol{x} = \widehat{w}_0 + \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i^T\ \boldsymbol{x} = \widehat{w}_0 + \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i\ , \boldsymbol{x})$$

- In the last step above we used the kernel trick to substitute $\boldsymbol{x}_i^T \boldsymbol{x}$ with $k(\boldsymbol{x_i}, \boldsymbol{x})$.
- The kernel trick is needed to prevent underfitting (ensure the feature vector is sufficient rich that a linear classifier can separate the data).
- The sparsity and large margin are needed to prevent overfitting.

- Schoelkopf, B. and A. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

SVM for Classification

- We have seen that the negative log-likelihood for the logistic regression model is

$$L_{nll}(y, \eta) = -logp(y|\boldsymbol{x}, \boldsymbol{w}) = \log(1 + e^{-y\eta})$$

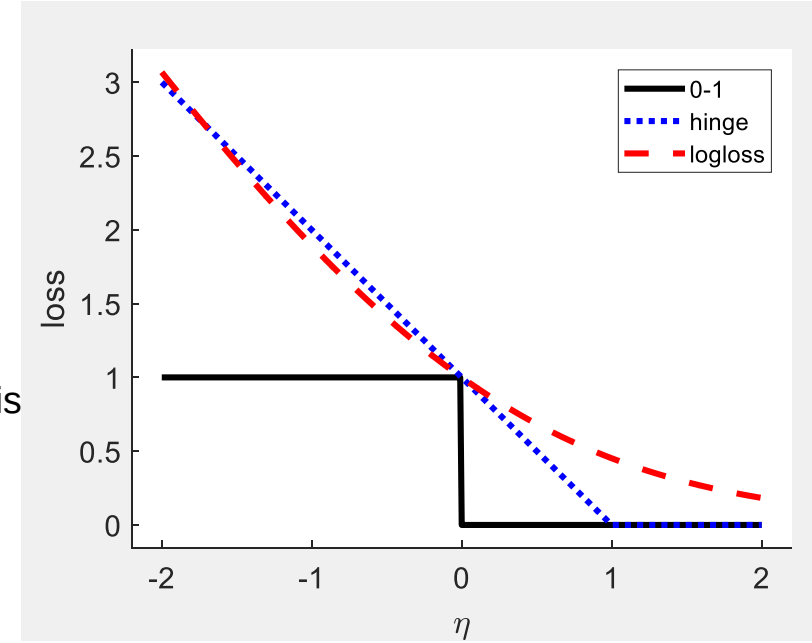
- It is a convex upper bound on the $0-1$ risk of a binary classifier, where $\eta = f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$ is the log odds ratio, and we have assumed the labels are $y \in \{1, -1\}$ rather than $\{0, 1\}$.

- We can replace the NLL loss with the **hinge loss**

$$L_{hinge}(y, \eta) = \max(0, 1 - y\eta) = (1 - y\eta)_+$$

The horizontal axis is the margin $y\eta$, the vertical axis is the loss. The log loss uses log base 2.

hingeLossPlot.m
from PMTK3



SVM for Classification

- Our optimization problem is as follows:

$$\min_{\boldsymbol{w},w_0} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(1 - y_i f(\boldsymbol{x}_i))_+$$

- Introducing slack variables we can write this as a quadratic program in $N + D + 1$ variables:

$$\min_{\boldsymbol{w},w_0,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \qquad \xi_i \geq 0, y_i\left(\boldsymbol{x}_i^T\boldsymbol{w} + w_0\right) \geq 1 - \xi_i, i = 1,..N$$

- We can eliminate $\boldsymbol{w}$, w_0 and ξ_i, and just solve the N dual variables, which correspond to the Lagrange multipliers for the constraints. Standard solvers take $\mathcal{O}(N^3)$ time. **Sequential minimal optimization** takes $\mathcal{O}(N^2)$. Linear SVMs, take $\mathcal{O}(N)$ time to train.

- Platt, J. (1998). Using analytic QP and sparseness to speed training of support vector machines. In *NIPS*.
- Joachims, T. (2006). Training Linear SVMs in Linear Time. In *Proc. Of the Int'l Conf. on Knowledge Discovery and Data Mining*.
- Bottou, L., O. Chapelle, D. DeCoste, and J. Weston (Eds.) (2007). *Large Scale Kernel Machines*. MIT Press.

SVMs for Classification

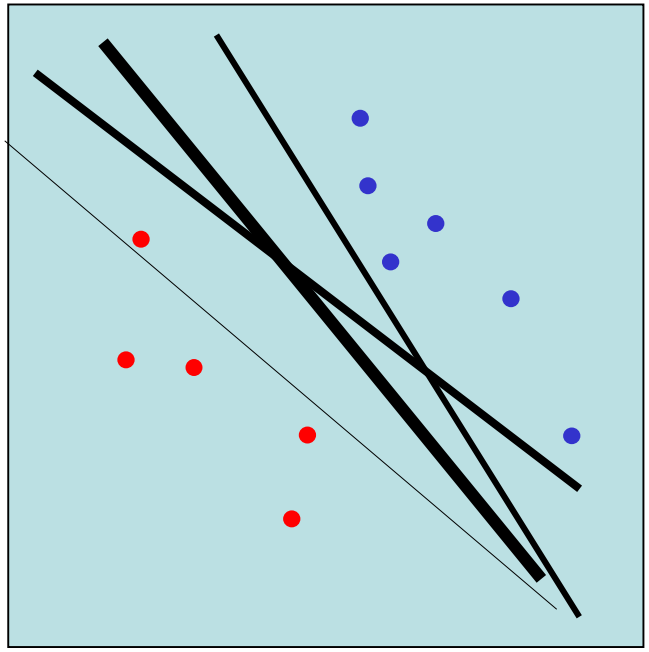
- One can show that: $\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i,\ \alpha_i = \lambda_i y_i, \alpha\ is\ sparse$
- The $\boldsymbol{x}_i$'s for which $\alpha_i > 0$ are called support vectors – these are points which are incorrectly classified or are classified correctly but are on or inside the margin.
- Making predictions:

$$\hat{y}(\boldsymbol{x}) = sgn(\hat{w}_0 + \hat{\boldsymbol{w}}^T \boldsymbol{x}) = sgn(\hat{w}_0 + \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i\ ,\ \boldsymbol{x}))$$

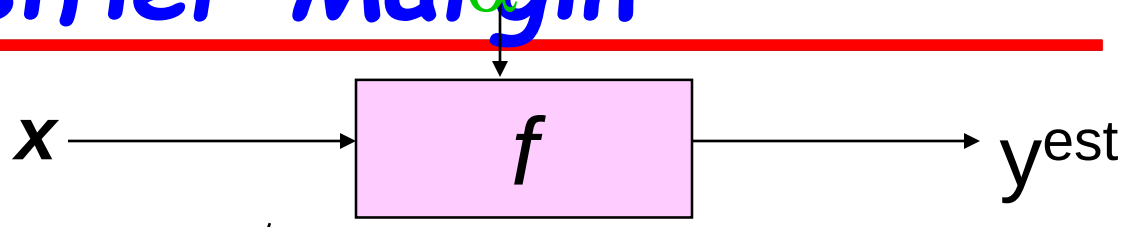
- It takes take $\mathcal{O}(sD)$ time to compute where $s \leq N$ is the number of support vectors.

Linear Classifiers

- Suppose we use a big set of features to ensure that the two classes are linearly separable. What is the best separating line to use?
  - ✓ The Bayesian answer is to use them all (including ones that do not quite separate the data.)
  - ✓ Weight each line by its posterior probability (i.e. by a combination of how well it fits the data and how well it fits the prior).
  - ✓ Is there an efficient way to approximate the correct Bayesian answer?

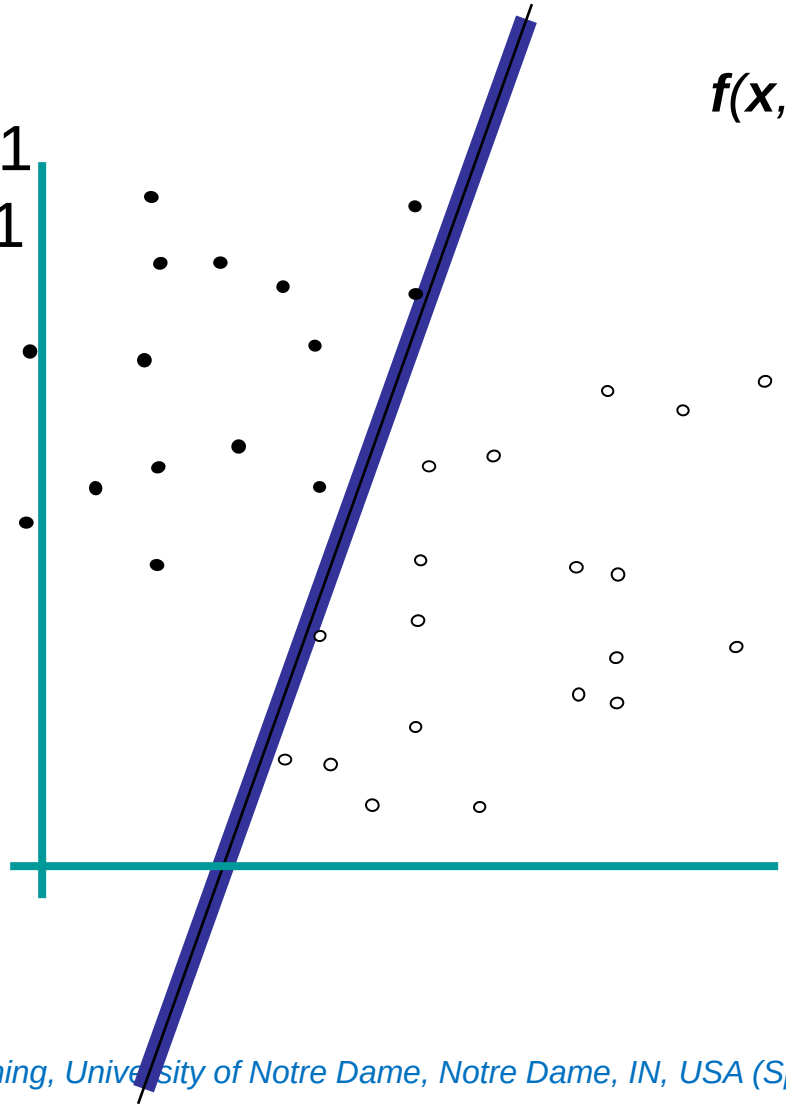


Classifier Margin



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\ \boldsymbol{x} - b)$

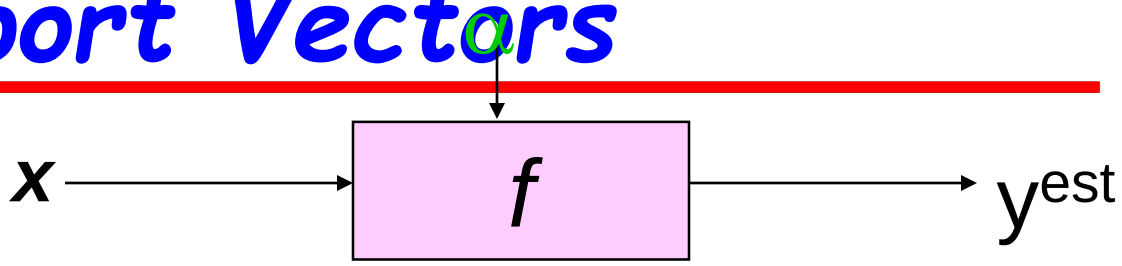
- denotes +1
- denotes -1



Linear SVM

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Support Vectors



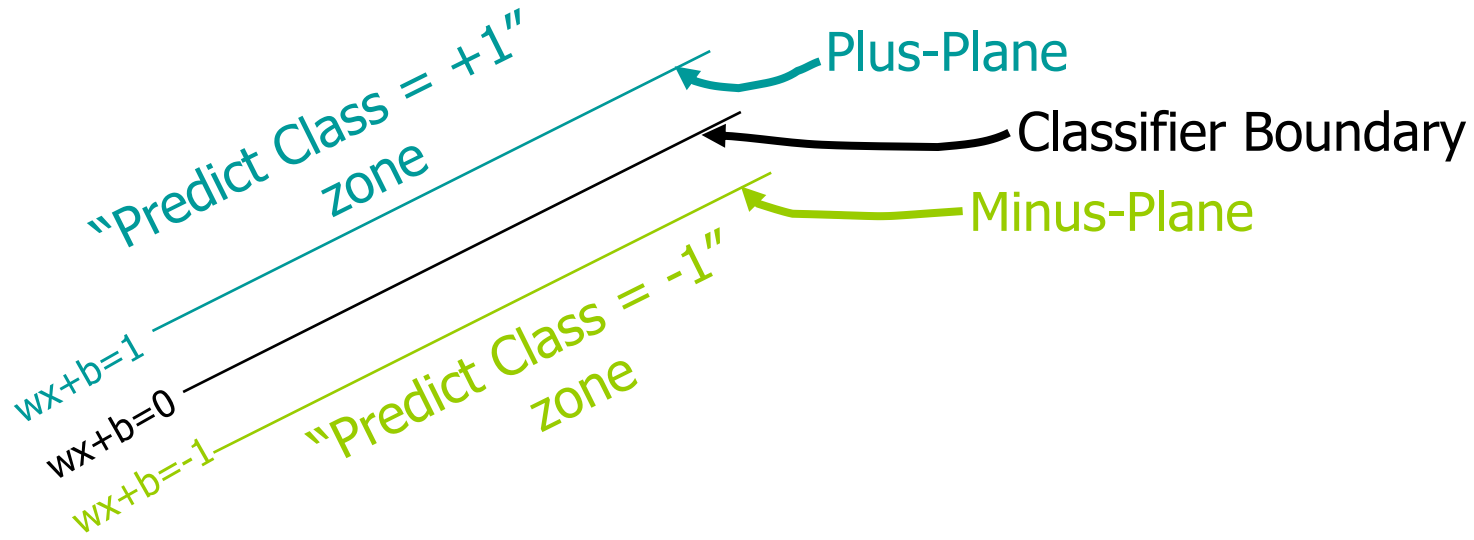
$f(x,w,b) = sign(w.\ x - b)$

- denotes +1
- denotes -1

Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the maximum margin.

Specifying a line and margin

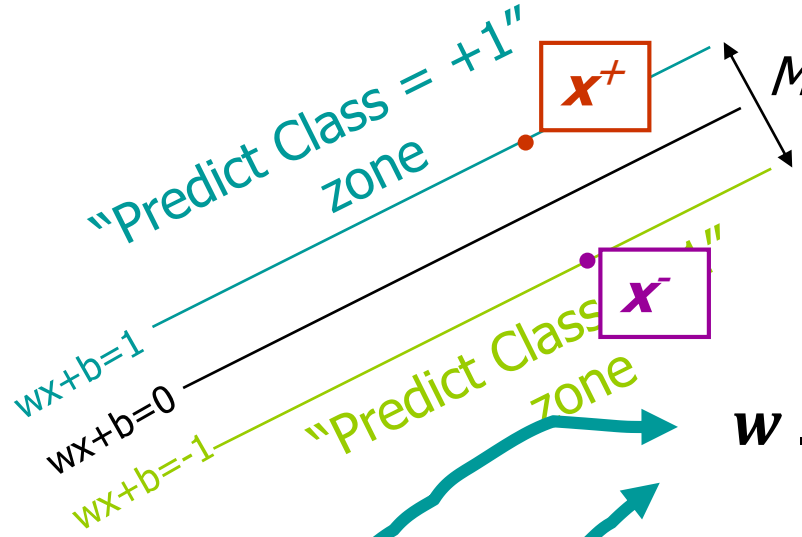


- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$

Classify as.. +1 if $\boldsymbol{w} \cdot \boldsymbol{x} + b >= 1$

-1 if $\boldsymbol{w} \cdot \boldsymbol{x} + b <= -1$

Computing the margin width



What we know:

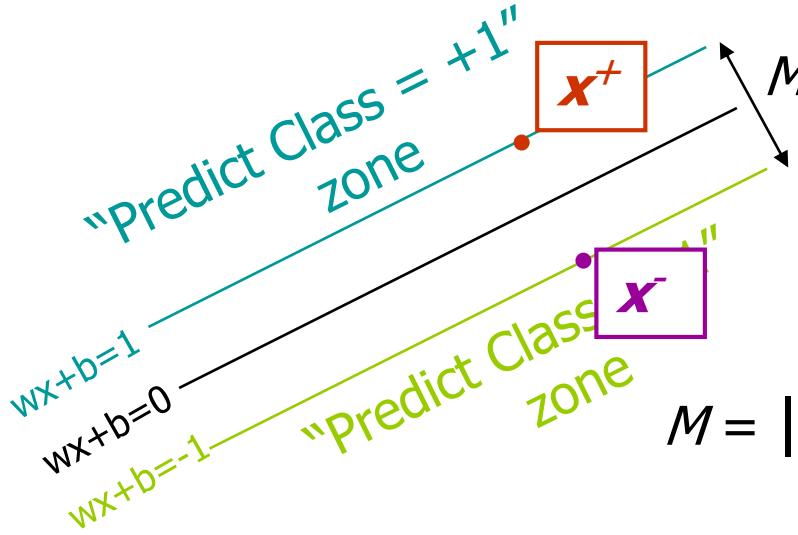
- $\boldsymbol{w} . \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} . \boldsymbol{x}^- + b = -1$
- $\boldsymbol{x}^+ = \boldsymbol{x}^- + \lambda \boldsymbol{w}$
- $|\boldsymbol{x}^+ - \boldsymbol{x}^-| = M$

It's now easy to get M in terms of $\boldsymbol{w}$ and b

$$\boldsymbol{w} . (\boldsymbol{x}^- + \lambda \boldsymbol{w}) + b = 1$$
$$=>$$
$$\boldsymbol{w} . \boldsymbol{x}^- + b + \lambda \boldsymbol{w} . \boldsymbol{w} = 1$$
$$=>$$
$$-1 + \lambda \boldsymbol{w} . \boldsymbol{w} = 1$$
$$=>$$

$$\lambda = \frac{2}{\mathbf{w}.\mathbf{w}}$$

Computing the margin width



M = Margin Width = $\frac{2}{\sqrt{\mathbf{w}.\mathbf{w}}}$

What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
- $|\mathbf{x}^+ - \mathbf{x}^-| = M$
-

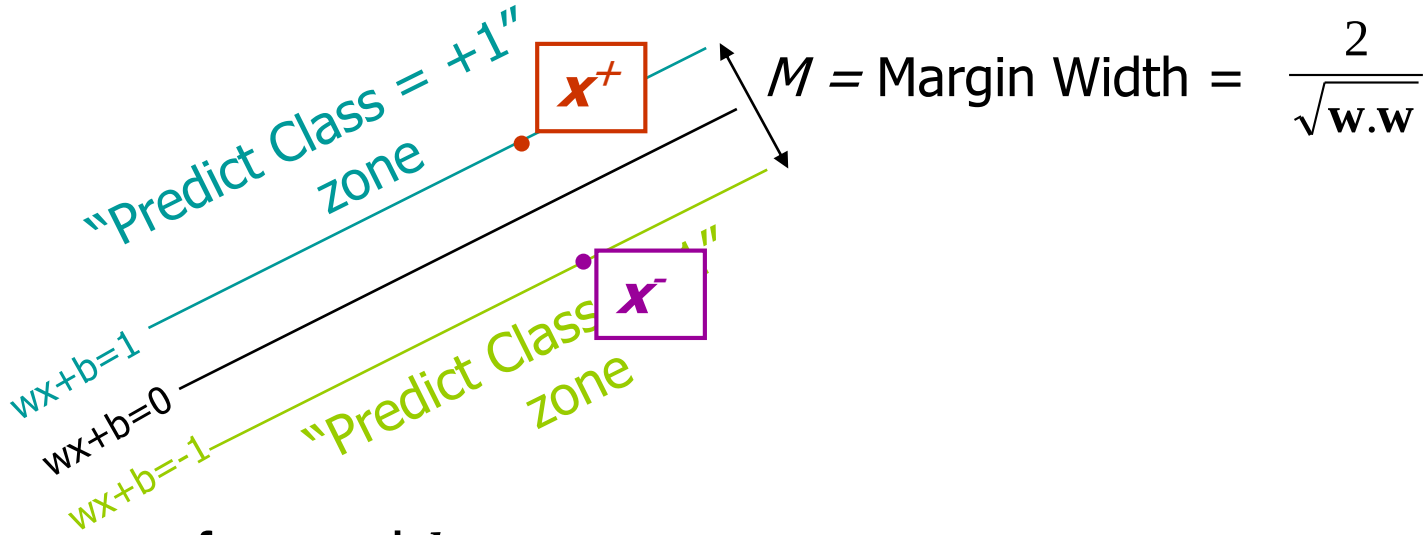
$$\lambda = \frac{2}{\mathbf{w}.\mathbf{w}}$$

$$M = |\mathbf{x}^+ - \mathbf{x}^-| = |\lambda \mathbf{w}| =$$

$$= \lambda |\mathbf{w}| = \lambda \sqrt{\mathbf{w}.\mathbf{w}}$$

$$= \frac{2\sqrt{\mathbf{w}.\mathbf{w}}}{\mathbf{w}.\mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}.\mathbf{w}}}$$

Learning the Maximum Margin Classifier



Given a guess of $\boldsymbol{w}$ and b we can

- Compute whether all data points in the correct half-planes
- Compute the width of the margin

So now we just need to write a program to search the space of $\boldsymbol{w}$'s and b's to find the widest margin that matches all the datapoints. *How?*

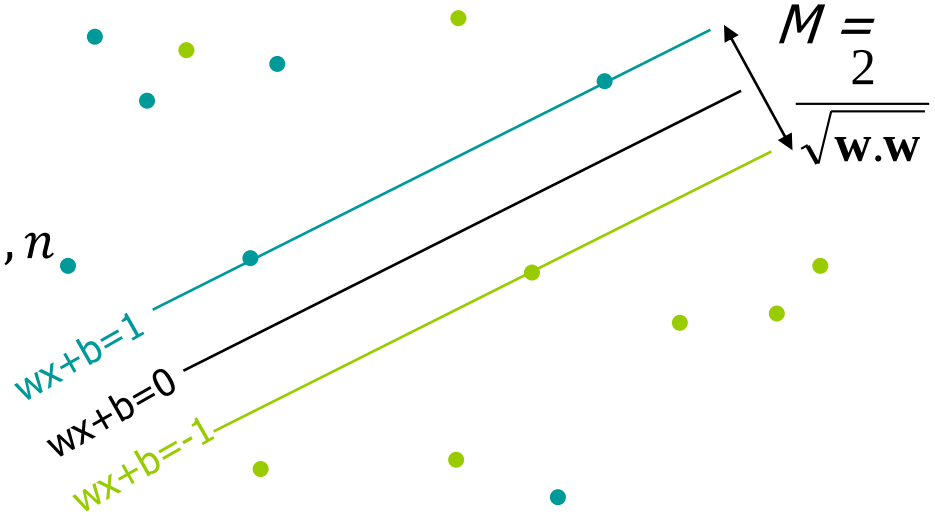
Support Vector Machine

- When the training examples are linearly separable we can maximize a geometric notion of margin (distance to the boundary) by minimizing the regularization penalty

$$\frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\sum_{i=1}^{D} w_i^2$$

subject to the classification constraints

$$y_i[w_0 + \boldsymbol{x}_i^T \boldsymbol{w}] - 1 \geq 0, i = 1, \ldots, n$$



- The solution is defined only on the basis of a subset of examples or "support vectors"

SVM for the non-separable case

- When the examples are not linearly separable we can modify the optimization problem slightly to add a penalty for violating the classification constraints: We minimize

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

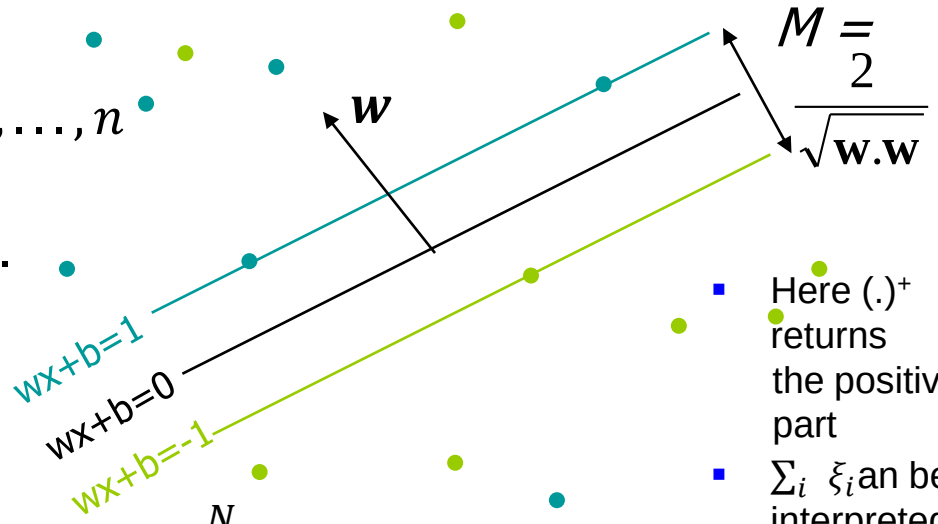
subject to the classification constraints

$$y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right] - 1 + \xi_i \geq 0, i = 1,\ldots,n$$

- Here $\xi_i \geq 0$ are slack variables.

- Can write the minimization objective as:

$$C\sum_{i=1}^{N}\left(1 - y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^+ + \frac{1}{2}\|\boldsymbol{w}\|^2 = C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\|\boldsymbol{w}\|^2$$



- Here $(.)^+$ returns the positive part
- $\sum_i \xi_i$ an be interpreted as the number of misclassified points

SVM for the non-separable case

- Note that the minimization objective

$$C\sum_{i=1}^{N}\left(1-y_i\left[w_0+\boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^+ +\frac{1}{2}\|\boldsymbol{w}\|^2$$

  is similar to that based on logistic regression.

- The corresponding regularized loss takes the form:

$$\frac{1}{N}\sum_{i=1}^{N}\left(1-y_i\left[w_0+\boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^+ +\frac{1}{NC}\|\boldsymbol{w}\|^2,$$

- Here the regularization parameter is $\lambda=\frac{1}{NC}$

- The first term above corresponds to the negative log likelihood:

$$\frac{1}{N}\sum_{i=1}^{N}-logg\left(y_i\left[w_0+\boldsymbol{x}_i^T\boldsymbol{w}\right]\right)\ ,g(z)=\frac{1}{1+e^{-z}}=logistic\ function$$

Probabilistic Interpretation of SVM

- The corresponding regularized loss takes the form:

$$\frac{1}{N}\sum_{i=1}^{N}\left(1-y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^+ + \frac{1}{NC}\|\boldsymbol{w}\|^2,$$

- One in the pursue of a probabilistic interpretation of SVM can show:

$$exp\left(-2\left(1-y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^+\right)$$
$$= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_i}} exp\left(-\frac{1}{2}\frac{\left(1+\lambda_i-y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right)^2}{\lambda_i}\right) d\lambda_i$$

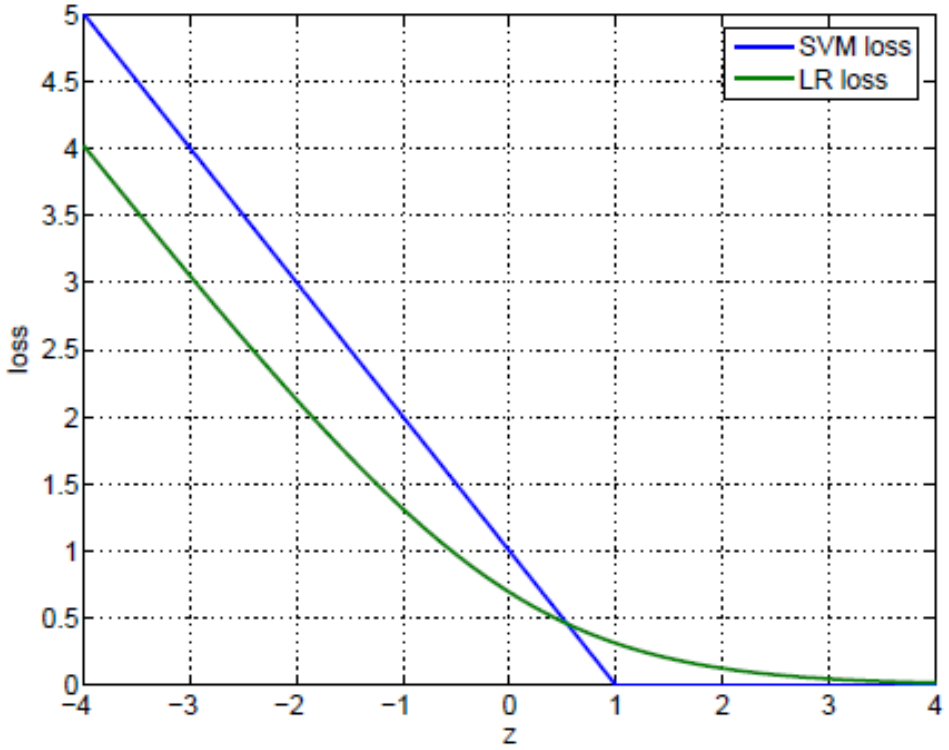
- This allows to fit the SVM with EM using λ_i as latent variables.

- Sollich, P. (2002). Bayesian methods for support vector machines: evidence and predictive class probabilities. *Machine Learning 46*, 21– 52.
- Polson, N. and S. Scott (2011). Data augmentation for support vector machines. *Bayesian Analysis 6*(1), 1–124.
- Franc, V., A. Zien, and B. Schoelkopf (2011). Support vector machines as probabilistic models. In *Intl. Conf. on Machine Learning*.

SVM Vs. Logistic Regression

- Common objective the minimization of:

$$\frac{1}{N}\sum_{i=1}^{N} Loss\left(y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right) + \frac{1}{NC}\|\boldsymbol{w}\|^2,$$



SVM Loss: $Loss(z) = (1-z)^+$

LR Loss: $Loss(z) = \frac{1}{1+e^{-z}}$

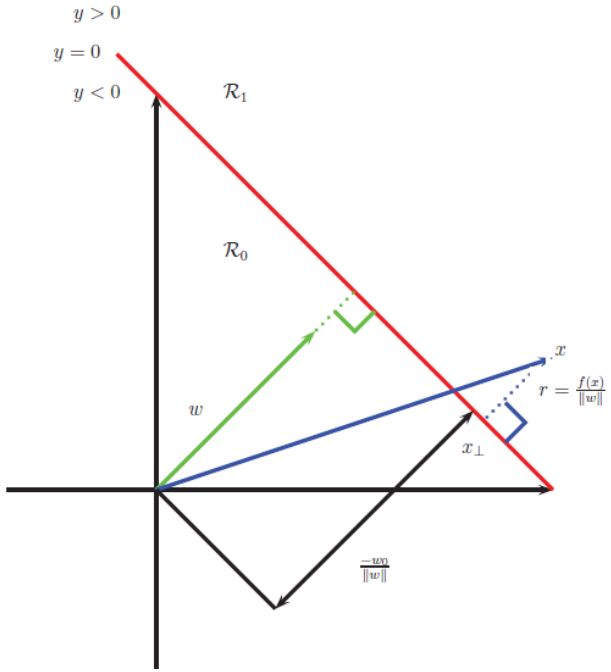
SVM – Separable Case

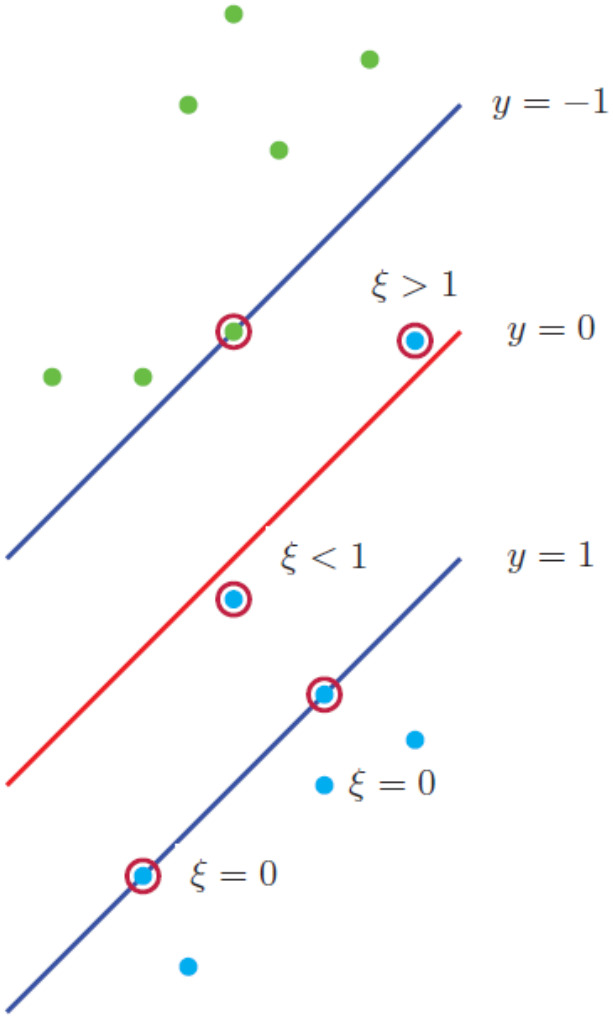
- Our problem is stated as follows:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 \; subject\ to$$

$$y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right] - 1 \geq 0, i = 1, \ldots, N$$

(each point is on the correct side of the boundary)





SVM – Separable Case

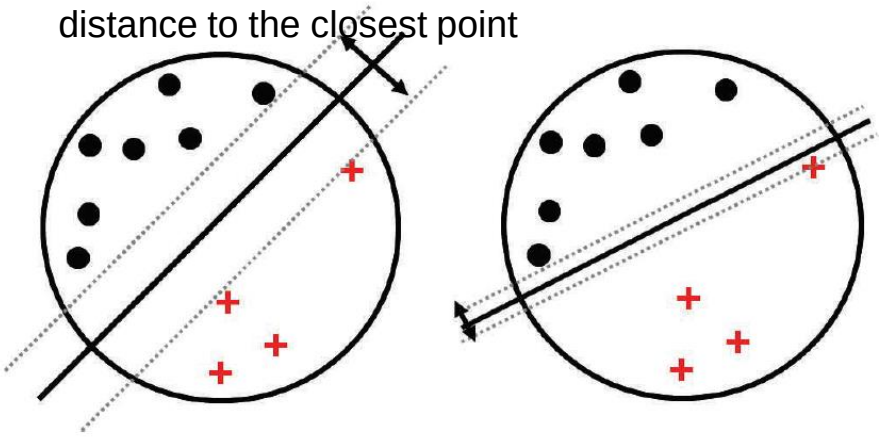
- Our problem is stated as follows:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 \; subject\ to$$

$$y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right] - 1 \geq 0, i = 1, \ldots, N$$

(each point is on the correct side of the boundary)

Margin here defined as the perpendicular distance to the closest point



Large margin Small margin

- We represent the constraints with Lagrange multipliers as losses:

$$\max_{\alpha \geq 0} \alpha_i \left(1 - y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right) = \begin{cases} 0, if\, y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right] - 1 \geq 0 \\ \infty, \qquad otherwise \end{cases}$$

- The minimization problem can now be stated as:

$$\min_{\boldsymbol{w}} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \max_{\alpha \geq 0} \alpha_i \left(1 - y_i\left[w_0 + \boldsymbol{x}_i^T\boldsymbol{w}\right]\right) \right\}$$

SVM – Separable Case

- The minimization problem can now be stated as:

$$\min_{\boldsymbol{w}} \max_{\{\alpha_i \geq 0\}} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left(1 - y_i \left[w_0 + \boldsymbol{x}_i^T \boldsymbol{w}\right]\right) \right\} =$$

$$\max_{\{\alpha_i \geq 0\}} \min_{\boldsymbol{w}} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left(1 - y_i \left[w_0 + \boldsymbol{x}_i^T \boldsymbol{w}\right]\right) \right\}$$

- As a result we should be able to minimize wrt $\boldsymbol{w}$ for any given set of the α's the following:

$$J(\boldsymbol{w}; \alpha) = \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left(1 - y_i \left[w_0 + \boldsymbol{x}_i^T \boldsymbol{w}\right]\right) \right\}$$

- Setting the derivatives wrt to $\boldsymbol{w}$ and w_0 gives:

$$\boldsymbol{w} - \sum_{i=1}^{N} \alpha_i \, y_i \boldsymbol{x}_i = 0, \; - \sum_{i=1}^{N} \alpha_i \, y_i = 0$$

SVM – Separable Case

- Substituting these Eqs $\widehat{\boldsymbol{w}} - \sum_{i=1}^{N} \alpha_i \, y_i \boldsymbol{x}_i = 0, \; -\sum_{i=1}^{N} \alpha_i \, y_i = 0$ into the objective function

$$\max_{\{\alpha_i \geq 0\}} \min_{\boldsymbol{w}} \left\{ \frac{1}{2} \|\widehat{\boldsymbol{w}}\|^2 + \sum_{i=1}^{N} \alpha_i \left(1 - y_i \left[w_0 \; + \; \boldsymbol{x}_i^T \widehat{\boldsymbol{w}}\right]\right) \right\}$$

gives:

$$\max_{\{\alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0\}} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j \right\}$$

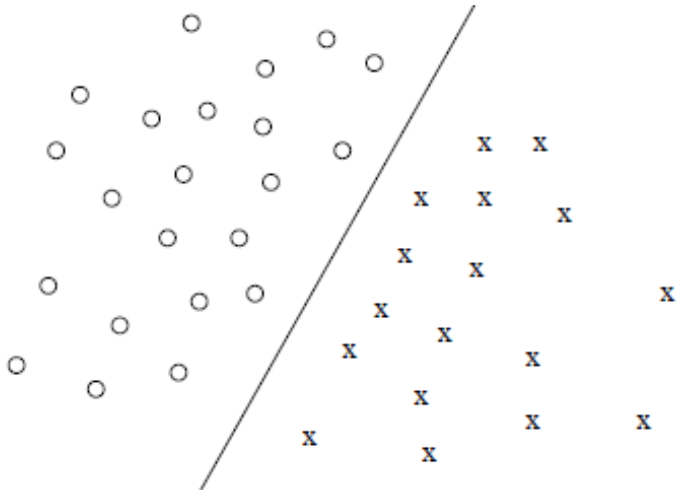
- Only $\alpha_i' s$ corresponding to "support vectors" will be non-zero.

- Once this optimization problem is solved, can do predictions on any test $\boldsymbol{x}$ according to the sign of the discriminant function

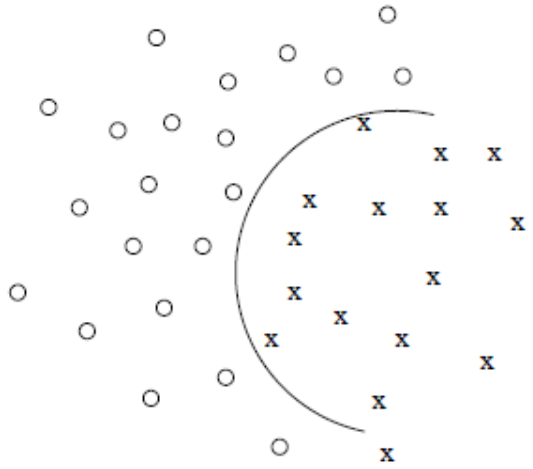
$$\widehat{w}_0 + \boldsymbol{x}^T \boldsymbol{w} = \widehat{w}_0 + \boldsymbol{x}^T \sum_{i,j=1}^{N} \hat{\alpha}_i y_i \boldsymbol{x}_i \; = \widehat{w}_0 + \sum_{i \in SV} \hat{\alpha}_i y_i \left(\boldsymbol{x}^T \, \boldsymbol{x}_i\right)$$

Non-linear Classifier

- So far our classifier can make only linear separations.
- We can easily obtain a non-linear classifier by using the kernel trick.
  Map $\boldsymbol{x} = [x_1\ x_2]$ into $\boldsymbol{\phi}(\boldsymbol{x}) = \left[x_1^2\ x_2^2\ \sqrt{2}x_1x_2\ \sqrt{2}x_1\ \sqrt{2}x_2\ 1\right]$



Linear separator in the feature ϕ-space



Non-linear separator in the original x-space

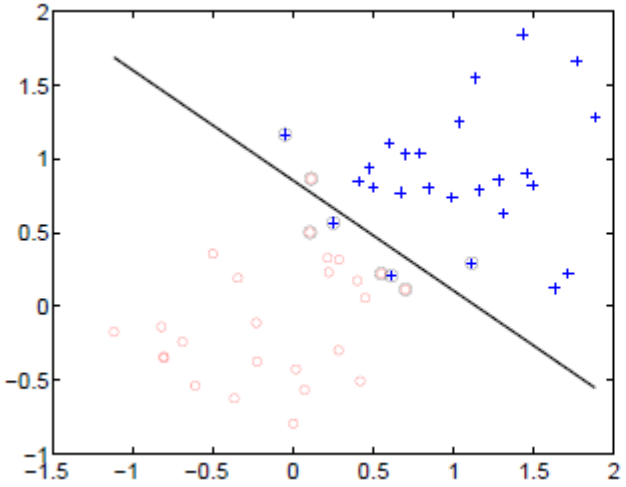
- The algorithm only needs dot products of the form:

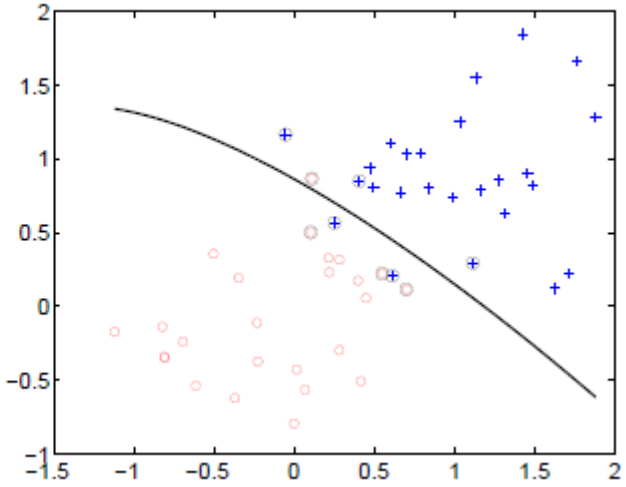
$$\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}') = x_1^2x_1'^2 + x_2^2x_2'^2 + 2x_1x_2x_1'x_2' + 2x_1\ x_1' + 2x_2\ x_2' + 1 = (1 + \boldsymbol{x}^T\boldsymbol{x}')^2$$

- So the inner products can be evaluated without ever explicitly constructing the feature vectors $\boldsymbol{\phi}(\boldsymbol{x})$!

SVM

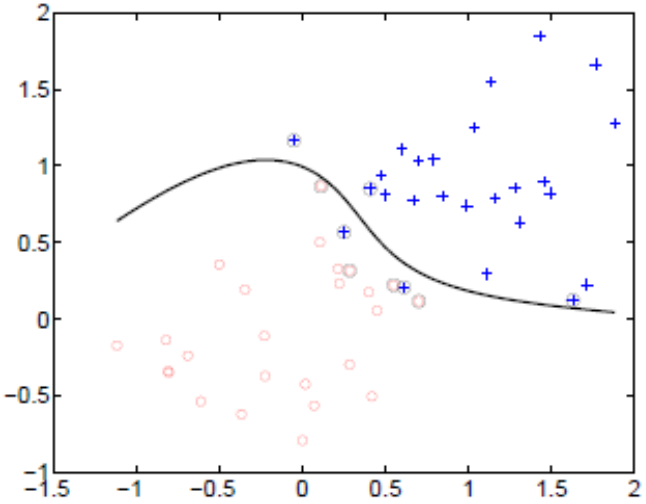
linear

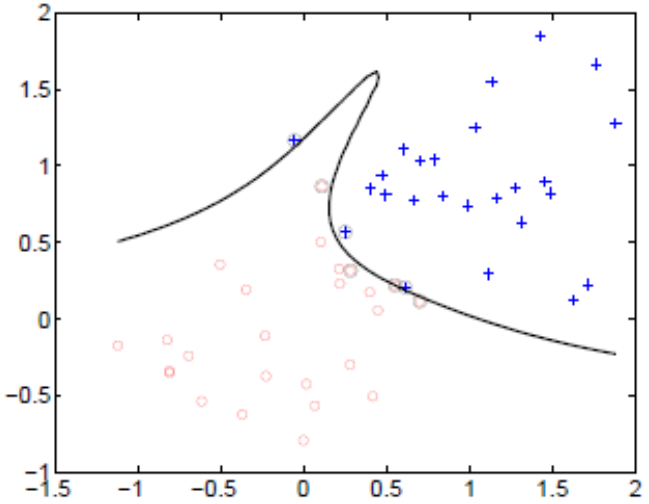




2nd order polynomial

4th order polynomial

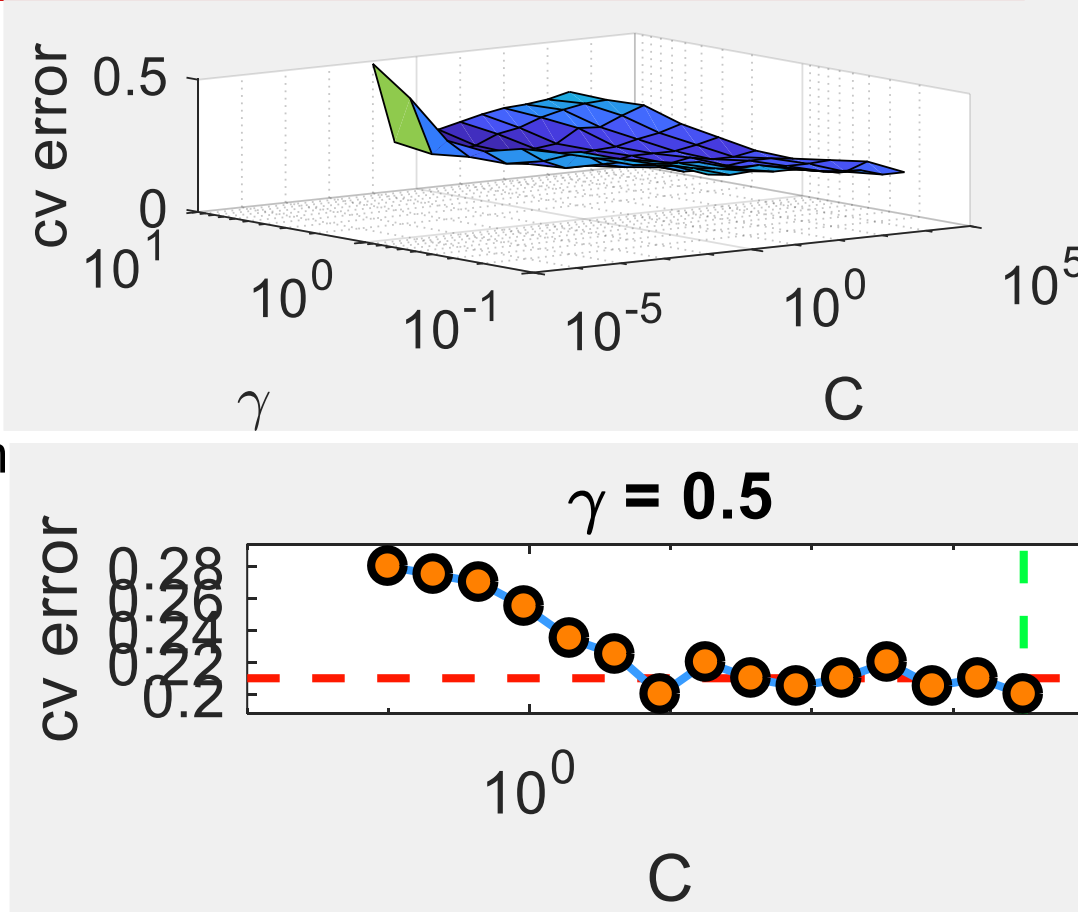




8th order polynomial

Choosing C

- Need to specify the kernel parameters and C. Typically C is chosen by cross-validation.
- C interacts quite strongly with the kernel parameters.
- Consider an RBF kernel with precision $\gamma = \frac{1}{(2\sigma)^2}$
- If γ = 5, corresponding to narrow kernels, we need heavy regularization, and hence small C (so λ = 1/C is big).
- If γ = 1, a larger value of C should be used.



CV estimate of the 0-1 risk as a function of C and γ.

svmCGammaDemo.m
from PMTK3

Choosing C

- libsvm recommends using CV over a 2d grid with values $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.
- Standardize the data first, for a spherical Gaussian kernel to make sense.
- To choose C efficiently, one can develop a path following the spirit of lars.
  - The basic idea is to start with λ large, so that the margin 1/||w(λ)|| is wide, and hence all points are inside of it and have $\alpha_i = 1$.
  - By slowly decreasing λ, a small set of points will move from inside the margin to outside, and their α_i values will change from 1 to 0, as they cease to be support vectors.
  - When λ is maximal, the function is completely smoothed, and no support vectors remain.

- Hastie, T., S. Rosset, R. Tibshirani, and J. Zhu (2004). The entire regularization path for the support vector machine. *J. of Machine Learning Research 5*, 1391–1415.

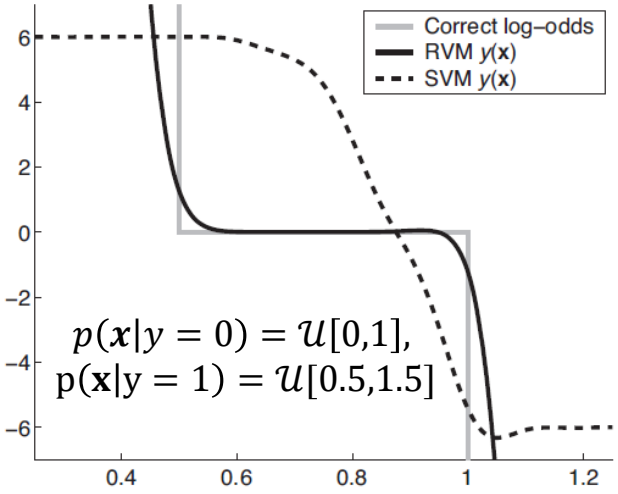
Probabilistic Response

- An SVM classifier produces a hard-labeling, $\hat{y}(\boldsymbol{x}) = sign(f(\boldsymbol{x}))$.

- A heuristic approach to producing confidence in our prediction is to interpret $f(\boldsymbol{x})$ as the log-odds ratio, $\log\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}$ and convert the output of an SVM to a probability using

$$p(y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \sigma(af(\boldsymbol{x}) + b)$$

where a, b can be estimated by maximum likelihood on a separate validation set (using the training set to estimate a and b leads to severe overfitting.)



- The log-odds of class 1 over class 0 should be zero in [0.5, 1.0], and infinite outside this region.
- We sampled 1000 points from the model, and then fit an RVM and an SVM with a Gaussian kenel of width 0.1.
- Both models can perfectly capture the decision boundary, and achieve a generalizaton error of 25%, which is Bayes optimal in this problem.
- The probabilistic output from the RVM is a good approximation to the true log-odds, but this is not the case for the SVM.

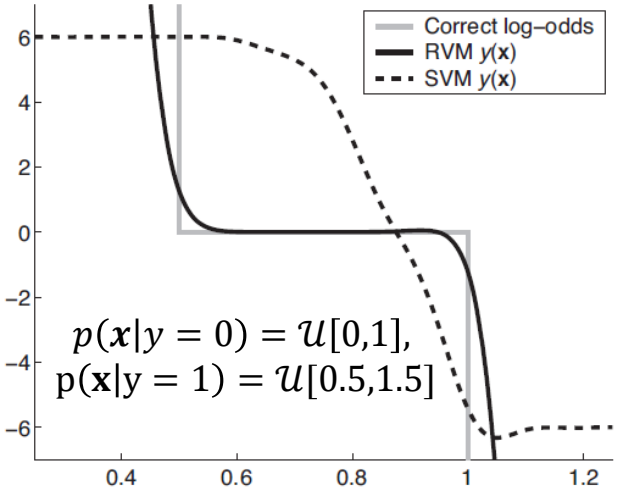
Probabilistic Response

- An SVM classifier produces a hard-labeling, $\hat{y}(\boldsymbol{x}) = sign(f(\boldsymbol{x}))$.

- A heuristic approach to producing confidence in our prediction is to interpret $f(\boldsymbol{x})$ as the log-odds ratio, $\log\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}$ and convert the output of an SVM to a probability using

$$p(y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \sigma(af(\boldsymbol{x}) + b)$$

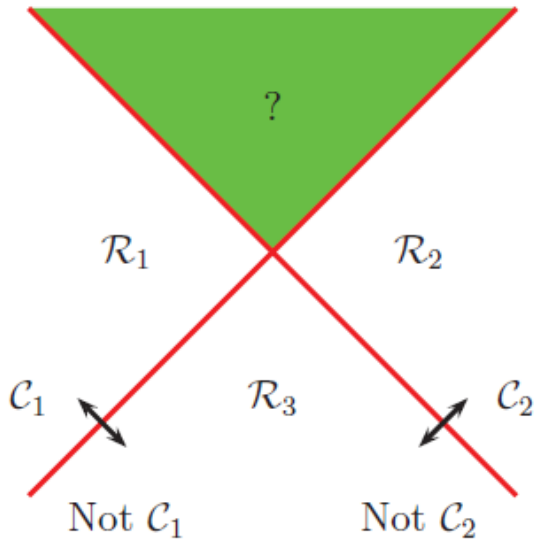
where a, b can be estimated by maximum likelihood on a separate validation set (using the training set to estimate a and b leads to severe overfitting.)



- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244.

Multiclass Classification

- The obvious approach is to use a **one-vs-all** approach. We train C binary classifiers, $f_c(x)$, where the data from class c is treated as positive, and the data from all the other classes as negative.
- This can result in ambiguously labeled regions.
- Can pick $\hat{y}(\boldsymbol{x}) = \underset{c}{\operatorname{argmax}} f_c(\boldsymbol{x})$. This will not work since the f_c functions don't have comparable magnitudes.

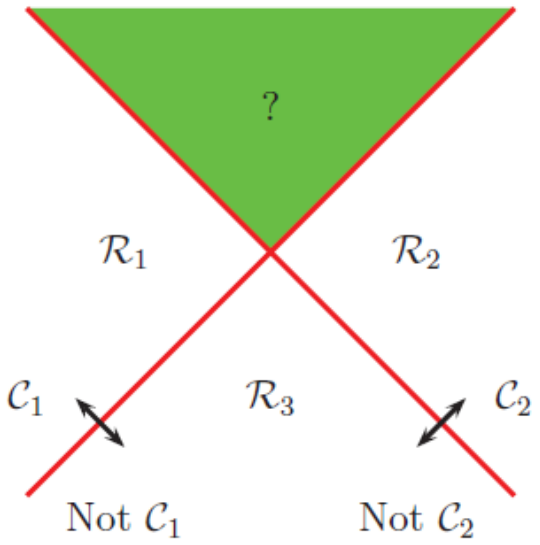


The one-versus-rest approach. The green region is predicted to be both class 1 and class 2.

- Weston, J. and C. Watkins (1999). Multi-class support vector machines. In *ESANN*.

Multiclass Classification

- Each binary subproblem is likely to suffer from the **class imbalance** problem.
- E.g. consider 10 equally represented classes. When training f_1, we will have 10% positive examples and 90% negative examples, which can hurt performance.
- One can train all C classifiers simultaneousy, but the resulting method takes $\mathcal{O}(C^2N^2)$ time, instead of $\mathcal{O}(CN^2)$.

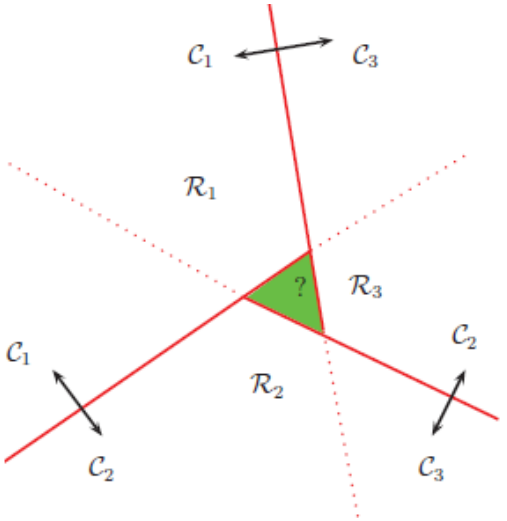


The one-versus-rest approach. The green region is predicted to be both class 1 and class 2.

- Weston, J. and C. Watkins (1999). Multi-class support vector machines. In *ESANN*.

Multiclass Classification

- Another approach is the **one-versus-one.**
- We train $C(C-1)/2$ classifiers to discriminate all pairs $f_{c,c'}$. We then classify a point into the class which has the highest number of votes.
- This can also result in ambiguities.
- It takes $\mathcal{O}(C^2N^2)$ time to train and $\mathcal{O}(C^2N_{sv})$ to test each data point, where N_{sv} is the number of support vectors.2
- Fundamental limitation of SVMs: they do not model uncertainty, so their output scores are not comparable across classes..



The one-versus-one approach.

- Allwein, E., R. Schapire, and Y. Singer (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *J. of Machine Learning Research*, 113–141.

Relevance Vector Machines

Relevance Vector Machines

- Relevance Vector Machines (RVM) where introduced by M. Tipping in Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211–244.
- It is a *Bayesian regression* technique on a *generalized linear model* that produces *sparse representations* (in the sense that many of the coefficients of the generalized linear model turn out to be zero).
- We will first describe it on one output dimension and then generalize it to many dimensions.
- Finally, we will couple it with an adaptive input decomposition algorithm and apply it to uncertainty quantification tasks. See I. Bilionis and N. Zabaras. *Multidimensional adaptive relevance vector machines for uncertainty quantification*. SIAM Journal for Scientific Computing, Vol. 34, No. 6, pp. B881–B908 2012 for further details.

Relevance Vector Machines

- RVM addresses many of the limitations of SVM:
  - SVM outputs are decisions rather than posterior probabilities.
  - Extention to K>2 classes in SVM is problematic.
  - There are complexity parameters in SVM to be found by cross-validation.
  - Predictions in SVM are based on kernel functions centered at the training data that are required to be positive definite.

RVM: Likelihood

- We assume that we observe the data:

$$\mathcal{D} = \left\{\boldsymbol{x}^{(i)}, y^{(i)} = f\left(\boldsymbol{x}^{(i)}\right)\right\}_{i=1}^{n},$$

and that we wish to learn the function $f(\cdot)$.

- Pick a set of M basis functions $\boldsymbol{\phi}: \boldsymbol{R}^k \rightarrow \boldsymbol{R}^M$:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left(\phi_1(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})\right).$$

- We will approximate $f(\cdot)$ by:

$$\hat{f}(x; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + \epsilon = \sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}) + \epsilon,$$

where ϵ is a random variable representing noise.

- For simplicity, we choose to work with Gaussian noise with zero mean and inverse variance β, so the **likelihood** is:

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \beta^{-1})$$

RVM: Prior

- The essence of RVM is the following **prior on the weights**:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} p\left(w_j|\alpha_j\right),$$

with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ and:

$$p\left(w_j|\alpha_j\right) = \mathcal{N}\left(w_j|0, \alpha_j^{-1}\right).$$

- The characteristic of this prior is that:

$$\lim_{\alpha_j \to \infty} p\left(w_j|\alpha_j\right) = \delta\left(w_j\right).$$

- This means that the j-th basis function can be removed from the model if its α_j is very big. This is because, its weights will be very sharply picked about zero.

- To complete the model, we also need to assign a prior $p(\beta, \boldsymbol{\alpha})$. We manage to get rid of it by employing the *evidence approximation.*

RVM: Posterior

- The **posterior of the weights** is:

$$p(\boldsymbol{w}|\mathcal{D},\beta,\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m},\boldsymbol{\Sigma}),$$

where:

$$\boldsymbol{\Sigma} = \left(\mathrm{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1},$$

$$\boldsymbol{m} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{t},$$

with $\boldsymbol{t} = \left(t^{(1)},\ldots,t^{(n)}\right)$ is the *vector of observations*, $\boldsymbol{\Phi} \in \boldsymbol{R}^{N\times M}$ is the *design matrix*:

$$\Phi_{ij} = \phi_j\left(\boldsymbol{x}^{(i)}\right),$$

and $\mathrm{diag}(\boldsymbol{\alpha})$ is a diagonal matrix with $\boldsymbol{\alpha}$ on the diagonal.

Proof:

First, notice that we can write:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} p\left(w_j|\alpha_j\right) = \prod_{j=1}^{M} \mathcal{N}\left(w_j|0,\alpha_j^{-1}\right) = \mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}).$$

RVM: Posterior

Proof (continuation):

Also, notice that the likelihood of the data (under an independence assumption) is given by:

$$p(\mathcal{D}|\boldsymbol{w},\beta)=\prod_{i=1}^{n}\mathcal{N}\left(t^{(i)}\middle|\boldsymbol{w}^T\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right),\beta^{-1}\right)=\mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{w},\beta^{-1}\boldsymbol{I}_N),$$

where $\boldsymbol{I}_N$ is the N-dimensional unit matrix and we have used the fact that:

$$\sum_{i=1}^{N}\|\,\boldsymbol{w}^T\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)-t^{(i)}\,\|^2=\sum_{i=1}^{N}\left(\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)\boldsymbol{w}-t^{(i)}\right)^T\left(\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)\boldsymbol{w}-t^{(i)}\right)$$
$$=(\boldsymbol{\Phi}\boldsymbol{w}-\mathbf{y})^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{w}\ -\mathbf{y}).$$

Finally, we have:

$$p(\boldsymbol{w}|\mathcal{D},\boldsymbol{\alpha},\beta)\propto p(\mathcal{D}|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{\alpha})=\mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{w},\beta^{-1}\boldsymbol{I}_N)\mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}).$$

RVM: Posterior

Proof (continuation):

Remember that the posterior is:

$$p(\boldsymbol{w}|\mathcal{D},\boldsymbol{\alpha},\beta) \propto p(\mathcal{D}|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\mathbf{w},\beta^{-1}\boldsymbol{I}_N)\mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}).$$

Now, we just have to look at it and complete the square…

You can simply drop this guy!

We have for the exponential inside the right hand side:

$$(\boldsymbol{\Phi}\boldsymbol{w}-\boldsymbol{t})^{\mathrm{T}}(\beta\boldsymbol{I}_N)(\boldsymbol{\Phi}\boldsymbol{w}-\boldsymbol{t})+\boldsymbol{w}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{w} =$$
$$\boldsymbol{w}^T\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\right)\boldsymbol{y} + \beta\boldsymbol{t}^T\boldsymbol{t} + \boldsymbol{w}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{w} =$$
$$\boldsymbol{w}^T\left(\mathrm{diag}(\boldsymbol{\alpha})+\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\right)\mathbf{t} =$$
$$\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\right)\mathbf{t}.$$

From which the result follows directly. End of Proof.

Important remark: When completing the square for a distribution, you can subtract, add or even ignore anything that does not depend on the random variable whose posterior you are calculating!

Or using Results for Linear Gaussian Models

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}, \Lambda^{-1}\right)$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}\left(\mathbf{y} \mid \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}\right)$$

- For the above linear Gaussian model, the following hold:

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y} \mid \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\Lambda^{-1}\mathbf{A}^T\right)$$

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{x} \mid \left(\Lambda + \mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\left(\Lambda\boldsymbol{\mu} + \mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b})\right), \left(\Lambda + \mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\right)$$

- For our problem $\boldsymbol{x} \to \boldsymbol{w}, \boldsymbol{\mu} \to \mathbf{0}, \boldsymbol{\Lambda} \to \text{diag}(\boldsymbol{\alpha}), \boldsymbol{y} \to \boldsymbol{t}, \boldsymbol{A} \to \boldsymbol{\Phi}, \boldsymbol{b} \to \mathbf{0}, \boldsymbol{L} = \boldsymbol{\beta}\boldsymbol{I}_N$

$$p(\boldsymbol{t}|\boldsymbol{w}) = \mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\mathbf{w}, \beta^{-1}\boldsymbol{I}_N), p(\boldsymbol{w}) = \mathcal{N}_M\left(\boldsymbol{w}|\text{diag}(\boldsymbol{\alpha})^{-1}\right)$$

and: $p(\boldsymbol{w}|\mathcal{D}, \beta, \boldsymbol{\alpha}) = \mathcal{N}\left(\boldsymbol{w}\middle|\beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{t}, \underbrace{\left(\text{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}}_{\boldsymbol{\Sigma}}\right)$

RVM: Predictive distribution

The **predictive** distribution is easily found as follows:

$$p(t|\boldsymbol{x},\boldsymbol{X},\boldsymbol{t},\boldsymbol{\alpha},\beta) = \int p(t|\boldsymbol{x},\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t},\boldsymbol{\alpha},\boldsymbol{\beta})d\boldsymbol{w}$$
$$= \mathcal{N}(t|\boldsymbol{m}^T\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\phi}(\boldsymbol{x}) + \beta^{-1}).$$

So, we have the *predictive mean (**m** being the posterior mean)*:

$$\boldsymbol{\mu}(\boldsymbol{x}) = \boldsymbol{m}^T\boldsymbol{\phi}(\boldsymbol{x}),$$

and the predictive variance:

$$\sigma^2(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\phi}(\boldsymbol{x}) + \beta^{-1}.$$

Proof:

Complete the square….

End of proof.

RVM: Evidence Approximation

- We will select the hyper-parameters $\boldsymbol{\alpha}$ and β by maximizing employing the *evidence approximation* (otherwise known as *maximizing the marginal likelihood*). We start by motivating it.

- Assume that we have a prior for $\boldsymbol{\alpha}$ and β, $p(\boldsymbol{\alpha}, \beta)$. Then the posterior of these parameters is:

$$p(\boldsymbol{\alpha}, \beta|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\alpha}, \beta)p(\boldsymbol{\alpha}, \beta) = \int p(\mathcal{D}|\boldsymbol{w}, \beta)p(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w}\, p(\boldsymbol{\alpha}, \beta).$$

Marginal Likelihood (or Evidence)

- Under the evidence approximation, we assume that:
  - $p(\boldsymbol{\alpha}, \beta)$ is relatively flat.
  - $p(\mathcal{D}|\boldsymbol{\alpha}, \beta)$ has very sharp maxima.

Then, we may obtain point estimates of the hyper-parameters by identifying these maxima.

RVM: Evidence

- To carry out the evidence approximation, we must obtain an analytic expression for the marginal likelihood. This is possible for RVM:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta) = \log\big(p(\mathcal{D}|\boldsymbol{\alpha}, \beta)\big) = -\frac{1}{2}\left[N\log(2\pi) + \log|\boldsymbol{C}| + \boldsymbol{t}^T\boldsymbol{C}^{-1}\boldsymbol{t}\right],$$

where $\boldsymbol{C} \in \boldsymbol{R}^{N\times N}$, is given by:

$$\boldsymbol{C} = \beta^{-1}\boldsymbol{I}_N + \boldsymbol{\Phi}\mathrm{diag}(\boldsymbol{\alpha})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}.$$

Proof:

This does require some more work that we avoided doing before. We start again from:

$$p(\mathcal{D}|\boldsymbol{w}, \beta)p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}_N(\boldsymbol{t}|\boldsymbol{\Phi}\mathbf{w}, \beta^{-1}\boldsymbol{I}_N)\mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}),$$

and we will complete the square making sure this time that we keep track of all terms that depend on $\boldsymbol{t}$.

RVM: Evidence

Proof (continuation):

Remember, that we want to integrate $\boldsymbol{w}$ out of:

$$p(\mathcal{D}|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}_n(\boldsymbol{t}|\boldsymbol{\Phi}\mathbf{w},\beta^{-1}\boldsymbol{I}_N)\mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}).$$

Again, we look at whatever is insider the exponential of the right-hand side and attempt to complete the square. This time, we already know the answer, but we need to keep track of $\boldsymbol{t}$ explicitly:

$$(\boldsymbol{\Phi}\boldsymbol{w}-\boldsymbol{t})^{\mathrm{T}}(\beta\boldsymbol{I}_N)(\boldsymbol{\Phi}\boldsymbol{w}-\boldsymbol{t}) + \boldsymbol{w}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{w} =$$
$$\boldsymbol{w}^T\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\right)\boldsymbol{t} + \beta\boldsymbol{t}^T\boldsymbol{t} + \boldsymbol{w}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{w} =$$
$$\boldsymbol{w}^T\left(\mathrm{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\right)\boldsymbol{t} + \beta\boldsymbol{t}^T\mathbf{t} =$$
$$\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\right)\boldsymbol{t} + \beta\boldsymbol{t}^T\boldsymbol{t} =$$
$$\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m} - \boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m} + \beta\boldsymbol{t}^T\boldsymbol{t} =$$
$$(\boldsymbol{w}-\boldsymbol{m})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{m}) + (\beta\boldsymbol{t}^T\boldsymbol{t} - \boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m}).$$

RVM: Evidence

Proof (continuation):

We can now write:

$$p(\mathcal{D}|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\mathbf{w},\beta^{-1}\boldsymbol{I}_N)\mathcal{N}_M(\boldsymbol{w}|\mathrm{diag}(\boldsymbol{\alpha})^{-1}) =$$
$$(2\pi)^{-\frac{N}{2}}\beta^{\frac{N}{2}}(2\pi)^{-\frac{M}{2}}|\mathrm{diag}(\boldsymbol{\alpha})|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{m})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{m})\right.$$

RVM: Two useful matrix identities

- *Woodbury Matrix Identity:*

$$(\boldsymbol{A}+\boldsymbol{UCV})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}.$$

- *Matrix Determinant Lemma:*

$$|\boldsymbol{A}+\boldsymbol{UWV}^{T}|=|\boldsymbol{W}^{-1}+\boldsymbol{V}^{T}\boldsymbol{A}^{-1}\boldsymbol{U}||\boldsymbol{W}||\boldsymbol{A}|.$$

You do not have to remember these, but you must be aware of there existence. Sometimes, they are the only route to analytic progress!

Or using Results for Linear Gaussian Models

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}, \Lambda^{-1}\right)$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}\left(\mathbf{y} \mid \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}\right)$$

- For the above linear Gaussian model, the following hold:

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y} \mid \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\Lambda^{-1}\mathbf{A}^T\right)$$

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{x} \mid \left(\Lambda + \mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\left(\Lambda\boldsymbol{\mu} + \mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b})\right), \left(\Lambda + \mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\right)$$

- For our problem $\boldsymbol{x} \to \boldsymbol{w}, \boldsymbol{\mu} \to \mathbf{0}, \boldsymbol{\Lambda} \to \text{diag}(\boldsymbol{\alpha}), \boldsymbol{y} \to \boldsymbol{t}, \boldsymbol{A} \to \boldsymbol{\Phi}, \boldsymbol{b} \to \mathbf{0}, \boldsymbol{L} = \boldsymbol{\beta}\boldsymbol{I}_N$

$$p(\boldsymbol{t}|\boldsymbol{w}) = \mathcal{N}_N(\boldsymbol{y}|\boldsymbol{\Phi}\mathbf{w}, \beta^{-1}\boldsymbol{I}_N), p(\boldsymbol{w}) = \mathcal{N}_M\left(\boldsymbol{w}|\text{diag}(\boldsymbol{\alpha})^{-1}\right)$$

and: $p(\boldsymbol{t}|\beta, \boldsymbol{\alpha}) = \mathcal{N}\left(\boldsymbol{w}\middle|\mathbf{0}, \beta^{-1}\boldsymbol{I}_N + \boldsymbol{\Phi}\text{diag}(\boldsymbol{\alpha})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\right)$

RVM: Evidence Approximation

- From our earlier derivation:

$$p(\mathcal{D}|\boldsymbol{\alpha},\beta)=(2\pi)^{-\frac{N}{2}}\beta^{\frac{N}{2}}|\text{diag}(\boldsymbol{\alpha})|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\boldsymbol{t}^T(\beta\boldsymbol{I}_N-\beta^2\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T)\boldsymbol{t}\right\}|\boldsymbol{\Sigma}|^{\frac{1}{2}}$$

where $\boldsymbol{\Sigma}=\left(\text{diag}(\boldsymbol{\alpha})+\beta\boldsymbol{\Phi}^{\text{T}}\boldsymbol{\Phi}\right)^{-1},\ \boldsymbol{m}=\beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\text{T}}\boldsymbol{t}$

- Noting that $\boldsymbol{t}^T\beta^2\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{t}=\boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m}$, can simplify as:

$$p(\mathcal{D}|\boldsymbol{\alpha},\beta)=(2\pi)^{-\frac{N}{2}}\beta^{\frac{N}{2}}|\text{diag}(\boldsymbol{\alpha})|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\beta\boldsymbol{t}^T\boldsymbol{t}-\boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m}\right)\right\}|\boldsymbol{\Sigma}|^{\frac{1}{2}}$$

- Taking derivatives wrt α_i of $\ln p(\mathcal{D}|\boldsymbol{\alpha},\beta)$ and using $\frac{\partial \ln|\Sigma|}{\partial\alpha_i}=Tr\left(\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial\alpha_i}\right)=-Tr\left(\boldsymbol{\Sigma}\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial\alpha_i}\right)=-\Sigma_{ii}$, $\boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{m}}{\partial\alpha_i}=-\boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\beta\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial\alpha_i}\boldsymbol{\Phi}^{\text{T}}\boldsymbol{t}=-m_i^2$ and setting equal to 0:

$$\frac{1}{2\alpha_i}+\frac{1}{2}m_i^2-m_i^2-\frac{1}{2}\Sigma_{ii}=0\rightarrow\alpha_i=\frac{1-\alpha_i\Sigma_{ii}}{m_i^2}\equiv\frac{\gamma_i}{m_i^2}$$

RVM: Evidence Approximation

$$p(\mathcal{D}|\boldsymbol{\alpha},\beta) = (2\pi)^{-\frac{N}{2}}\beta^{\frac{N}{2}}|\text{diag}(\boldsymbol{\alpha})|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\beta\boldsymbol{t}^T\boldsymbol{t} - \boldsymbol{m}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{m}\right)\right\}|\boldsymbol{\Sigma}|^{\frac{1}{2}}$$

- Taking derivatives wrt β of $\ln p(\mathcal{D}|\boldsymbol{\alpha},\beta)$ and using $\frac{\partial \ln|\Sigma|}{\partial\beta} = Tr\left(\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial\beta}\right) = -Tr\left(\boldsymbol{\Sigma}\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial\beta}\right) = -Tr\left(\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)$, and $\boldsymbol{m} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{t}$, setting equal to 0:

$$\frac{N}{2\beta} - \frac{1}{2}\boldsymbol{t}^T\boldsymbol{t} + \frac{1}{2}\boldsymbol{m}^T\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{m} + \boldsymbol{m}^T\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{t} - \frac{1}{2}tr\left(\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right) = 0 \rightarrow \left(\beta^{(new)}\right)^{-1} = \frac{\|\boldsymbol{t}-\boldsymbol{\Phi}\boldsymbol{m}\|^2}{N-\sum\gamma_i}$$

- Here, we used

$$\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} = \boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \boldsymbol{\beta}^{-1}\boldsymbol{\Sigma}\text{diag}(\boldsymbol{\alpha}) - \beta^{-1}\boldsymbol{\Sigma}\text{diag}(\boldsymbol{\alpha}) =$$
$$= \boldsymbol{\Sigma}\left(\text{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\beta^{-1} - \beta^{-1}\boldsymbol{\Sigma}\text{diag}(\boldsymbol{\alpha})$$
$$= \left(\text{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\left(\text{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\beta^{-1} - \beta^{-1}\boldsymbol{\Sigma}\text{diag}(\boldsymbol{\alpha})$$
$$= \beta^{-1}(\boldsymbol{I} - \text{diag}(\boldsymbol{\alpha})\boldsymbol{\Sigma})$$

and thus $tr\left(\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right) = \beta^{-1}\sum\gamma_i$

RVM: Evidence Approximation

$$\alpha_i = \frac{1 - \alpha_i \Sigma_{ii}}{m_i^2} \equiv \frac{\gamma_i}{m_i^2}, \quad \left(\beta^{(new)}\right)^{-1} = \frac{\|\boldsymbol{t} - \boldsymbol{\Phi m}\|^2}{N - \sum \gamma_i}$$

- Learning therefore proceeds by choosing initial values for $\boldsymbol{\alpha}$ and β, evaluating the mean and covariance of the posterior using $\boldsymbol{\Sigma} = \left(\text{diag}(\boldsymbol{\alpha}) + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}\right)^{-1}$, $\boldsymbol{m} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{t}$, respectively, and then alternately re-estimating the hyperparameters, using the Eqs. above and iterating until a suitable convergence criterion is satisfied.

- The second approach is to use the EM algorithm. These two approaches to finding the values of the hyperparameters that maximize the evidence are formally equivalent. Numerically, however, it is found that the direct optimization approach gives somewhat faster convergence (Tipping, 2001).

RVM: Fast training algorithm

- We will develop a **fast algorithm** for **maximizing the evidence**.
- We follow closely the developments in *Tipping, M. E. and A. C. Faul (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In C. M. Bishop and B. J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, Jan 3-6*.
- There are some typos in the paper.
- We will achieve this by asking the following question:
- A Matlab implementation for the one dimensional case can be found here.

Keeping β fixed and given some values for $\boldsymbol{\alpha}$, what is the action on a single basis function that gives the maximum increase in evidence?

This will require a very thorough examination of the evidence function.

RVM: Splitting the evidence

- Some notation:
  - Subscript "$-i$": Wherever we see this applied to a vector or matrix, it gives the corresponding vector or matrix with the influence of the i −th basis function removed.
  - $\boldsymbol{\alpha}_{-i} = (\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_m)$: All the hyper-parameters except the ones that pertain to basis function i.
  - $\boldsymbol{\Phi}_{-i}$: The design matrix with the column corresponding to basis function i removed.
  - $\boldsymbol{C}_{-i} = \beta^{-1}\boldsymbol{I}_N + \boldsymbol{\Phi}_{-\mathrm{i}}\mathrm{diag}(\boldsymbol{\alpha}_{-i})^{-1}\boldsymbol{\Phi}_{-i}^{\mathrm{T}}$: The matrix $\boldsymbol{C}$, without the influence of basis function i.
  - $\mathcal{E}(\boldsymbol{\alpha}_{-i}, \beta) = -\frac{1}{2}\left[n\log(2\pi) + \log|\boldsymbol{C}_{-i}| + \boldsymbol{y}^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{y}\right]$: The evidence when the i −th basis function has been removed.

RVM: Splitting the evidence

Given any basis function $i \in \{1, \dots, M\}$, the evidence can be split as:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta) = \mathcal{E}(\boldsymbol{\alpha}_{-i}, \beta) + \epsilon(\alpha_i),$$

where

$$\epsilon(\alpha_i) = \frac{1}{2}\left(\log \alpha_i - \log(\alpha_i + s_i) + \frac{h_i^2}{\alpha_i + s_i}\right),$$

with

$$s_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{\varphi}_i \quad \text{and} \quad h_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{t}.$$

Remark: Everything that relates to basis function i is there!

Proof:

We start by splitting $\boldsymbol{C}$*:*

$$\boldsymbol{C} = \beta^{-1}\boldsymbol{I}_N + \boldsymbol{\Phi}\text{diag}(\boldsymbol{\alpha})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$$

$$= \beta^{-1}\boldsymbol{I}_N + (\boldsymbol{\Phi}_{-\mathrm{i}} \quad \boldsymbol{\varphi}_i)\begin{pmatrix} \text{diag}(\boldsymbol{\alpha}_{-i})^{-1} & \boldsymbol{0}_{(\mathrm{M}-1)\times 1} \\ \boldsymbol{0}_{1\times(M-1)} & \alpha_i^{-1} \end{pmatrix}\begin{pmatrix} \boldsymbol{\Phi}_{-i}^T \\ \boldsymbol{\varphi}_i^T \end{pmatrix}$$

$$= \beta^{-1}\boldsymbol{I}_N + \boldsymbol{\Phi}_{-\mathrm{i}}\text{diag}(\boldsymbol{\alpha}_{-i})^{-1}\boldsymbol{\Phi}_{-\mathrm{i}}^{\mathrm{T}} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T$$

$$= \boldsymbol{C}_{-i} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T.$$

RVM: Splitting the evidence

Proof (continuation):

Remember that the evidence is:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta) = -\frac{1}{2}[N\log(2\pi) + \log|\boldsymbol{C}| + \boldsymbol{t}^T\boldsymbol{C}^{-1}\boldsymbol{t}].$$

We will only make use of this:

$$\boldsymbol{C} = \boldsymbol{C}_{-i} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T,$$

and some identities.

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}.$$

From the Woodbury identity, we get:

$$\boldsymbol{C}^{-1} = \boldsymbol{C}_{-i}^{-1} - \boldsymbol{C}_{-i}^{-1}\boldsymbol{\varphi}_i\left(\alpha_i + \boldsymbol{\varphi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\varphi}_i\right)^{-1}\boldsymbol{\varphi}_i^T\boldsymbol{C}_{-i}^{-1}$$

$$= \boldsymbol{C}_{-i}^{-1} - \frac{\boldsymbol{C}_{-i}^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T\boldsymbol{C}_{-i}^{-1}}{\alpha_i + s_i}.$$

And we get:

$$\boldsymbol{t}^T\boldsymbol{C}^{-1}\boldsymbol{t} = \boldsymbol{t}^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{t} - \frac{\left(\boldsymbol{t}^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\varphi}_i\right)\left(\boldsymbol{\varphi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{t}\right)}{\alpha_i + s_i} = \boldsymbol{t}^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{t} + \frac{h_i^2}{\alpha_i + s_i}.$$

RVM: Splitting the evidence

Proof (continuation):

Finally, we look at $|\boldsymbol{C}|$. Remember:

$$\boldsymbol{C} = \boldsymbol{C}_{-i} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T,$$

Using the matrix determinant lemma:

$$|\boldsymbol{A} + \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T| = |\boldsymbol{W}^{-1} + \boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{U}||\boldsymbol{W}||\boldsymbol{A}|$$

$$|\boldsymbol{C}| = \left|\alpha_i + \boldsymbol{\varphi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\varphi}_i\right|\left|\alpha_i^{-1}\right||\boldsymbol{C}_{-i}|$$
$$= \frac{|\boldsymbol{C}_{-i}|(\alpha_i + s_i)}{\alpha_i}.$$

Or

$$\log|\boldsymbol{C}| = \log|\boldsymbol{C}_{-i}| - \log\alpha_i + \log(\alpha_i + s_i).$$

RVM: Splitting the evidence

Proof (continuation):

Now we combine everything. Remember that:

$$\boldsymbol{t}^T \boldsymbol{C}^{-1} \boldsymbol{t} = \boldsymbol{t}^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{t} + \frac{h_i^2}{\alpha_i + s_i},$$

and

$$\log|\boldsymbol{C}| = \log|\boldsymbol{C}_{-i}| - \log \alpha_i + \log(\alpha_i + s_i)\,.$$

Therefore, we have for the evidence:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta) = -\frac{1}{2}[N \log(2\pi) + \log|\boldsymbol{C}| + \boldsymbol{t}^T \boldsymbol{C}^{-1} \boldsymbol{t}]$$

$$= -\frac{1}{2}\left(N \log(2\pi) + \log|\boldsymbol{C}_{-i}| + \boldsymbol{t}^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{t} + \frac{h_i^2}{\alpha_i + s_i} - \log \alpha_i + \log(\alpha_i + s_i) \right)$$

$$= \mathcal{E}(\boldsymbol{\alpha}_{-i}, \beta) + \epsilon(\alpha_i).$$

End of proof.

RVM: Analysis of the evidence

We have shown that for any basis function $i \in \{1, \ldots, M\}$, the evidence can be split as:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta) = \mathcal{E}(\boldsymbol{\alpha}_{-i}, \beta) + \epsilon(\alpha_i),$$

where

$$\epsilon(\alpha_i) = \frac{1}{2}\left(\log \alpha_i - \log(\alpha_i + s_i) + \frac{h_i^2}{\alpha_i + s_i}\right).$$

We will now analyze the $\epsilon(\alpha_i)$ term in more detail.

We compute its first derivative:

$$\frac{\partial \epsilon(\alpha_i)}{\partial \alpha_i} = \frac{s_i^2 - \alpha_i \theta_i}{2\alpha_i(\alpha_i + s_i)^2},$$

where

$$\theta_i = h_i^2 - s_i.$$

RVM: Analysis of the evidence

Setting

$$\frac{\partial \epsilon(\alpha_i)}{\partial \alpha_i} = \frac{s_i^2 - \alpha_i \theta_i}{2\alpha_i(\alpha_i + s_i)^2}$$

equal to zero, we have two possibilities:

- If $\theta_i > 0$, then

$$\alpha_i^* = \frac{s_i^2}{\theta_i},$$

is the unique maximum of $\epsilon(\alpha_i)$ (calculate the 2nd derivative to be convinced that it is indeed a maximum).

- If $\theta_i \leq 0$, then

$$\alpha_i^* = +\infty,$$

Is the unique maximum of $\epsilon(\alpha_i)$ (convince yourselves that its derivative remains positive for all α_i). This means that removing the corresponding basis function from the model, increases the evidence.

RVM: Possible actions

The previous observations translate into three possible actions for each basis function i:

- **ADD**, if $\theta_i > 0$ and $\alpha_i = +\infty$:

$$\alpha_i^{\text{new}} = \frac{s_i^2}{\theta_i},$$
$$\Delta\mathcal{E} = \epsilon(\alpha_i^{\text{new}}).$$

- **RE-ESTIMATE**, if $\theta_i > 0$ and $\alpha_i < +\infty$:

$$\alpha_i^{\text{new}} = \frac{s_i^2}{\theta_i},$$
$$\Delta\mathcal{E} = \epsilon(\alpha_{\text{i}}^{\text{new}}) - \epsilon(\alpha_i).$$

- **DELETE**, if $\theta_i \leq 0$ and $\alpha_i < +\infty$:

$$\alpha_i^{\text{new}} = +\infty,$$
$$\Delta\mathcal{E} = -\epsilon(\alpha_i).$$

RVM: Updating the noise

Now, we assume that all the $\boldsymbol{\alpha}$'s are kept fixed and we attempt to maximize the evidence with respect to the inverse noise β.

Taking the derivative of the evidence with respect to β, you get:

$$\beta^{\text{new}} = \frac{N - M - \sum_i \alpha_i \Sigma_{ii}}{\| \boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu} \|^2}.$$

This is a lengthy calculation. You are going to need the following formulas for a general matrix $\boldsymbol{A}$:

$$\frac{d\boldsymbol{A}^{-1}}{dt} = -\boldsymbol{A}^{-1}\frac{d\boldsymbol{A}}{dt}\boldsymbol{A}^{-1},$$

$$\frac{d|\boldsymbol{A}|}{dt} = |\boldsymbol{A}|\text{tr}\left[\boldsymbol{A}^{-1}\frac{d\boldsymbol{A}}{dt}\right].$$

RVM: Complete Algorithm

1. Initialize β and $\boldsymbol{\alpha}$ (Some of them might be $+\infty$. The corresponding basis functions are not in the model).

2. Compute: $\boldsymbol{\Sigma}, \boldsymbol{m}$ only for the basis functions in the model but **all** the s_i and h_i's.

3. For $i = 1, \ldots, M$, compute $\theta_i = h_i^2 - s_i$ and find the basis function i^* and the corresponding action $\mathcal{A}_{i^*}$ that gives the maximum increase in evidence.

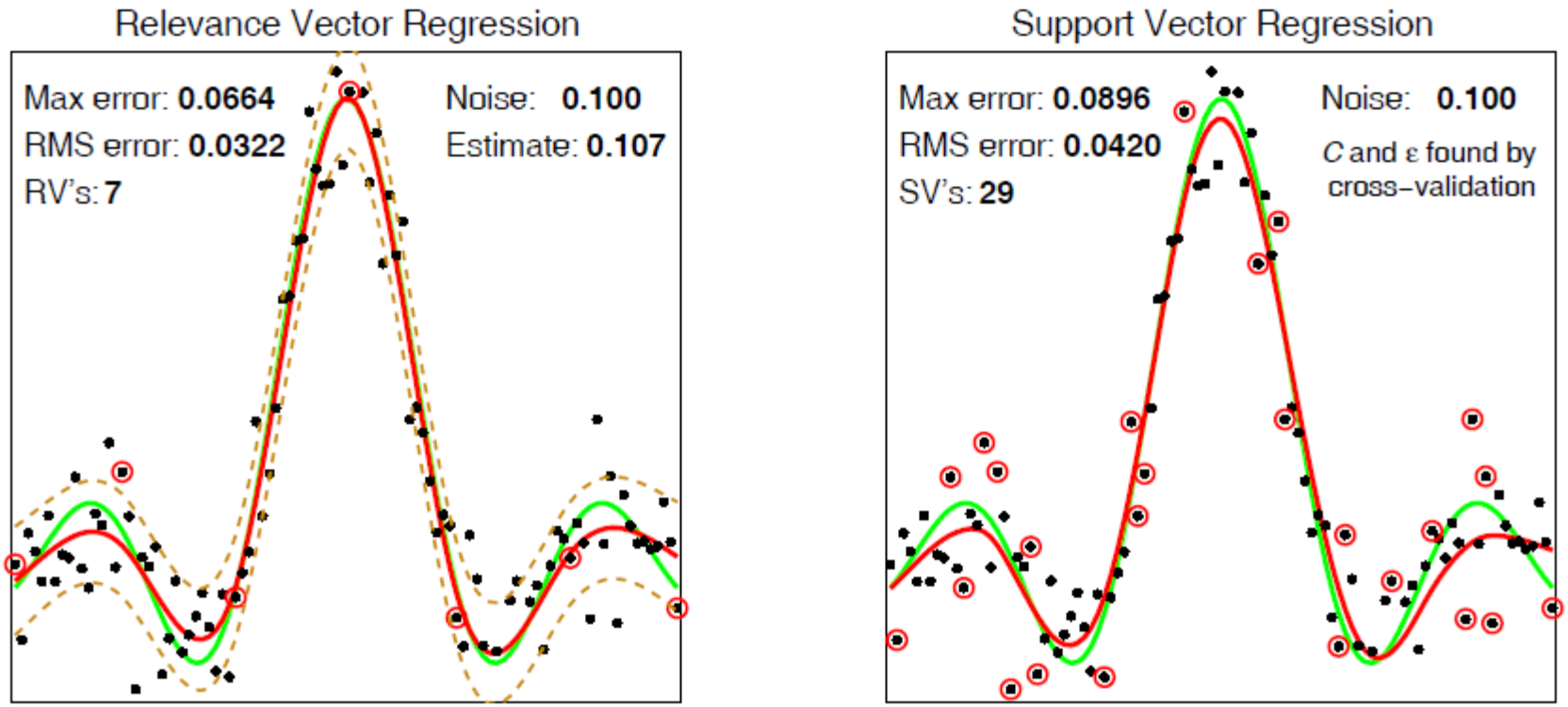
4. Perform the best action.

5. Re-estimate the noise β.

6. Re-compute (or update) $\boldsymbol{\Sigma}, \boldsymbol{m}, s_i$ and h_i.

7. If converged, stop. Otherwise go to 3.

A simple example



Fitting noisy samples from $f(x) = \frac{\sin(x)}{x}$. The basis functions are Gaussian kernels centered on the data points:

$$\phi_i(x) = k\left(x, x^{(i)}\right) = \exp\left(-\lambda\left(x - x^{(i)}\right)^2\right).$$

Taken from this very nice tutorial. The comparison is with SVM (a non-Bayesian competitor).

RVM: Fast updates

An efficient implementation of the RVM algorithm, requires a fast way to update the various statistics.

First, the mean and the co-variance matrix should only be build out of the **relevant** basis functions only:

$$\boldsymbol{\Sigma}_r = \left(\text{diag}(\boldsymbol{\alpha}_r) + \beta \boldsymbol{\Phi}_r^{\text{T}} \boldsymbol{\Phi}_r\right)^{-1},$$
$$\boldsymbol{m}_r = \beta \boldsymbol{\Sigma}_r \boldsymbol{\Phi}_r^{\text{T}} \boldsymbol{y},$$

where the subscript "r" denotes that the corresponding quantity corresponds only to the basis functions in the model (the relevant basis function, i.e. the ones with $\alpha_i < +\infty$. The other basis functions are simply irrelevant…

This requires some book-keeping. That is, at any iteration you must keep track of the set of relevant basis functions:

$$\mathcal{R} = \left\{j \in \{1, \ldots, m\} : a_j < +\infty\right\}.$$

RVM: Fast updates

Secondly, we come to the statistics:

$$s_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{\varphi}_i \quad \text{and} \quad h_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{t}.$$

These are required for **all** basis functions, so that the change in evidence can be computed. It is more convenient to keep track of the following statistics instead:

$$S_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}^{-1} \boldsymbol{\varphi}_i \quad \text{and} \quad \mathrm{H}_i = \boldsymbol{\varphi}_i^T \boldsymbol{C}^{-1} \boldsymbol{t}.$$

It is trivial to show that they connect with the original by:

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i} \quad \text{and} \quad h_i = \frac{\alpha_i H_i}{\alpha_i - S_i}.$$

Using the Woodbury identity once more, you can find an easy way to compute the capital statistics:

$$S_i = \beta \parallel \boldsymbol{\varphi}_i \parallel^2 - \beta^2 \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_r \boldsymbol{\Sigma}_r \boldsymbol{\Phi}_r^T \boldsymbol{\varphi}_i,$$
$$H_i = \beta \boldsymbol{\varphi}_i^T \boldsymbol{t} - \beta^2 \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_r \boldsymbol{\Sigma}_r \boldsymbol{\Phi}_r^T \boldsymbol{t}.$$

RVM: Fast updates

Finally, each action can be carried out without computing the statistics from scratch! You can derive them by repeatedly applying the Woodbury identity. It will take a couple of pages…

Notation:

- M_r: current number of relevant basis functions
- $i \in \{1, \ldots, M\}$: index of basis function that is currently being updated.
- $j \in \{1, \ldots, M_r\}$: the index within the **relevant** basis set that corresponds to i.
- $t \in \{1, \ldots, M\}$: index that runs over all basis functions (in or out of the model).

RVM: Fast updates

ADD basis function i:

$$\Delta\mathcal{E}^{\text{new}} = \frac{1}{2}\left(\frac{Q_i^2 - S_i}{S_i} + \log\frac{S_i}{Q_i^2}\right),$$

$$\boldsymbol{\Sigma}_r^{\text{new}} = \begin{pmatrix} \boldsymbol{\Sigma}_r + \beta^2\Sigma_{ii}\boldsymbol{\Sigma}_r\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_r\boldsymbol{\Sigma}_r & -\beta\Sigma_{ii}\boldsymbol{\Sigma}_r\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_i \\ -\beta\Sigma_{ii}\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_r\boldsymbol{\Sigma}_r & \boldsymbol{\Sigma}_{ii} \end{pmatrix},$$

$$\boldsymbol{m}_r^{\text{new}} = \begin{pmatrix} \boldsymbol{m}_r - \mu_i\beta\Sigma_{ii}\boldsymbol{\Sigma}_r\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_i \\ m_i \end{pmatrix},$$

$$S_t^{\text{new}} = S_t - \Sigma_{ii}(\beta\boldsymbol{\varphi}_t^T\boldsymbol{e}_i)^2,$$

$$H_t^{\text{new}} = H_t - \mu_i\beta\boldsymbol{\varphi}_t^T\boldsymbol{e}_i,$$

where

$$\Sigma_{ii} = (\alpha_i^{\text{new}} - S_i)^{-1},$$

$$\mu_i = \Sigma_{ii}H_i,$$

and

$$\boldsymbol{e}_i = \boldsymbol{\varphi}_i - \beta\boldsymbol{\Phi}_r\boldsymbol{\Sigma}_r\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_i.$$

Finally, m_r is increased by one.

CAUTION: There is a typo in Tipping and Faul.

RVM: Fast updates

DELETE basis function j:

$$\Delta\mathcal{E}^{\text{new}} = \frac{1}{2}\left(\frac{Q_i^2 - S_i}{S_i} - \log\left(1 - \frac{S_i}{\alpha_i}\right)\right),$$

$$\boldsymbol{\Sigma}_r^{\text{new}} = \boldsymbol{\Sigma}_r - \Sigma_{jj}^{-1}\boldsymbol{\Sigma}_j\boldsymbol{\Sigma}_j^T,$$

$$\boldsymbol{m}_r^{\text{new}} = \boldsymbol{m}_r - \mu_j\Sigma_{jj}^{-1}\boldsymbol{\Sigma}_j,$$

$$S_t^{\text{new}} = S_t + \Sigma_{\text{jj}}^{-1}\left(\beta\boldsymbol{\Sigma}_j^T\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_t\right)^2,$$

$$H_t^{\text{new}} = H_t + m_j\Sigma_{jj}^{-1}\beta\boldsymbol{\Sigma}_j^T\boldsymbol{\Phi}_r^T\boldsymbol{\varphi}_t,$$

where Σ_{jj} is the (j,j) element and $\boldsymbol{\Sigma}_j$ the j-th column of $\boldsymbol{\Sigma}_r$, resp., and m_j is the j-th element of $\boldsymbol{m}_r$.

After completing this step the corresponding rows and columns of the mean and the covariance must be removed.

The indices of the relevant vectors must be updated accordingly.

RVM: Fast updates

RE-ESTIMATE basis function j:

$$\Delta\mathcal{E}^{\text{new}} = \frac{1}{2}\left(\frac{Q_i^2}{S_i + \left(\left(\alpha_i^{\text{new}}\right)^{-1} - \alpha_i^{-1}\right)^{-1}} - \log\left(1 - S_i\left(\left(\alpha_i^{\text{new}}\right)^{-1} - \alpha_i^{-1}\right)\right)\right),$$

$$\boldsymbol{\Sigma}_r^{\text{new}} = \boldsymbol{\Sigma}_r - \kappa_j \boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_j^T,$$

$$\boldsymbol{m}_r^{\text{new}} = \boldsymbol{m}_r - \kappa_j m_j \boldsymbol{\Sigma}_j,$$

$$S_t^{\text{new}} = S_t + \kappa_j\left(\beta \boldsymbol{\Sigma}_j^T \boldsymbol{\Phi}_r^T \boldsymbol{\varphi}_t\right)^2,$$

$$H_t^{\text{new}} = H_t + \kappa_j m_j \beta \boldsymbol{\Sigma}_j^T \boldsymbol{\Phi}_r^T \boldsymbol{\varphi}_t,$$

$\boldsymbol{\Sigma}_j$ the j-th column of $\boldsymbol{\Sigma}_r$, resp., and μ_j is the j-th element of $\boldsymbol{m}_r$ and

$$\kappa_j = \left(\Sigma_{jj} + \left(\left(\alpha_i^{\text{new}}\right)^{-1} - \alpha_i^{-1}\right)^{-1}\right)^{-1},$$

where Σ_{jj} is the (j,j) element.

RVM: Extension to multiple outputs

- We assume that we observe the data:

$$\mathcal{D} = \left\{\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)} = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}\right)\right\}_{i=1}^{n},$$

and that we wish to learn the function $\boldsymbol{f}(\cdot) \in \boldsymbol{R}^q$.

- Pick a set of M basis functions $\boldsymbol{\phi}: \boldsymbol{R}^k \to \boldsymbol{R}^M$:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left(\phi_1(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})\right).$$

- We will approximate $\boldsymbol{f}(\cdot)$ by:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}; \boldsymbol{W}) = \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon}$ is a random vector representing noise and $\boldsymbol{W} \in \boldsymbol{R}^{M \times q}$.

- For simplicity, we choose to work with Gaussian noise with zero mean and inverse variance β, so the **likelihood** is:

$$p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{W}, \beta) = \mathcal{N}_q(\boldsymbol{t}|\boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}), \beta^{-1}).$$

RVM: Extension to multiple outputs

- The **prior on the weights**:

$$p(\boldsymbol{W}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} p(\boldsymbol{w}_j|\alpha_j),$$

with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ and:

$$p(\boldsymbol{w}_j|\alpha_j) = \mathcal{N}_q(\boldsymbol{w}_j|0, \alpha_j^{-1}).$$

- The **posterior of the weights** is:

$$p(\boldsymbol{W}|\mathcal{D}, \beta, \boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{W}|\boldsymbol{M}, \boldsymbol{\Sigma}),$$

where:

$$\boldsymbol{\Sigma} = \left(\text{diag}(\boldsymbol{\alpha}) + \beta \boldsymbol{\Phi}^{\text{T}} \boldsymbol{\Phi}\right)^{-1},$$
$$\boldsymbol{M} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\text{T}} \boldsymbol{Y},$$

with $\boldsymbol{Y} = \left(\boldsymbol{t}^{(1)}, \ldots, \boldsymbol{t}^{(N)}\right) \in \boldsymbol{R}^{N \times q}$ is the *vector of observations*, $\boldsymbol{\Phi} \in \boldsymbol{R}^{N \times M}$ is the *design matrix*:

RVM: Predictive distribution

- The **predictive** distribution is easily found as follows:

$$p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{\alpha},\beta) = \int p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{W},\beta)p(\boldsymbol{W}|\boldsymbol{\alpha})d\boldsymbol{W}$$
$$= \mathcal{N}(y|\boldsymbol{M}^T\boldsymbol{\phi}(\boldsymbol{x}),\boldsymbol{\varphi}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\varphi}(\boldsymbol{x}) + \beta^{-1}).$$

So, we have the *predictive mean*:

$$\mu(\boldsymbol{x}) = \boldsymbol{M}^T\boldsymbol{\phi}(\boldsymbol{x}),$$

and the predictive variance:

$$\sigma^2(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\varphi}(\boldsymbol{x}) + \beta^{-1}.$$

Proof:

Complete the square….

End of proof.

RVM: Extension to multiple outputs

- **REMARK:** You might have to work with a scaled version of the data so that the prior assumptions are closer to reality. You may attempt to fit the function:

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{L}_{\text{obs}}^{-1}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{m}_{\text{obs}}),$$

where $\boldsymbol{m}_{\text{obs}}$ is the *empirical mean of the data:*

$$\boldsymbol{m}_{\text{obs}} = \frac{1}{n}\sum_{i=1}^{N} \boldsymbol{t}^{(i)},$$

and $\boldsymbol{L}_{\text{obs}} \in \boldsymbol{R}^{q\times q}$ is the Cholesky decomposition of the *empirical covariance matrix of the data:*

$$\boldsymbol{C}_{\text{obs}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{t}^{(i)} - \boldsymbol{m}_{\text{obs}}\right)\left(\boldsymbol{t}^{(i)} - \boldsymbol{m}_{\text{obs}}\right)^{T}.$$

If the empirical covariance is rank-deficient, you may use either an incomplete Cholesky or the eigen-decomposition. This is essentially a sort of output dimensionality reduction.

RVM: Extension to multiple outputs

- We proceed again, by maximizing the total evidence.
- Working out the details, it is easy to show that the total evidence is the sum of the evidence of each output:

$$\mathcal{E}(\boldsymbol{\alpha}, \beta | \mathcal{D}) = \sum_{j=1}^{q} \mathcal{E}\left(\boldsymbol{\alpha}, \beta \middle| \mathcal{D}_j\right),$$

where $\mathcal{D}_j = \left\{\left(\boldsymbol{x}^{(i)}, t_j^{(i)}\right)\right\}_{i=1}^{N}$.

- The algorithmic details remain essentially the same. The only difference is that:

$$H_i = \frac{1}{q}\sum_{j=1}^{q} H_{ji},$$

$$H_{ji} = \beta \boldsymbol{\varphi}_i^T \boldsymbol{t}_j - \beta^2 \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_r \boldsymbol{\Sigma}_r \boldsymbol{\Phi}_r^T \boldsymbol{t}_j.$$

RVM: Numerical stability concerns

- There are several numerical instabilities that should concern you.
- This is very common.
- For a numerically stable way to implement RVM using the Generalized Singular Value Decomposition look at Appendix D of I. Bilionis and N. Zabaras. *Multidimensional adaptive relevance vector machines for uncertainty quantification*. SIAM Journal for Scientific Computing, Vol. 34, No. 6, pp. B881–B908 2012.

RVM For Classification

- We can extend the relevance vector machine framework to classification problems by applying the ARD prior over weights to a probabilistic linear classification model.
- Consider two-class problems with a binary target variable $t \in \{0, 1\}$.
- The model now takes the form of a linear combination of basis functions transformed by a logistic sigmoid function

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}))$$

- If we introduce a Gaussian prior over $\boldsymbol{w}$, then we obtain the model that has been considered in the logistic regression notes.

RVM For Classification

- In the RVM, this model uses the ARD prior

$$p(w_j|\alpha_j) = \mathcal{N}(w_j|0, \alpha_j^{-1}).$$

- We can no longer integrate analytically over $\boldsymbol{w}$. Here we follow Tipping (2001) and use the Laplace approximation.

- We begin by initializing $\boldsymbol{\alpha}$. For this given value of $\boldsymbol{\alpha}$, we then build a Gaussian approximation to the posterior distribution and thereby obtain an approximation to the marginal likelihood ($A = diag(\boldsymbol{\alpha_i})$)

$$lnp(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}) = ln\{p(\boldsymbol{t}|\boldsymbol{w})p(\boldsymbol{w}|\alpha)\} - lnp(\boldsymbol{t}|\boldsymbol{\alpha})$$

$$= \sum_{n=1}^{N} \{t_n lny_n + (1 - t_n)\ln(1 - y_n)\} - \frac{1}{2}\boldsymbol{w}^T \boldsymbol{A}\boldsymbol{w} + const.$$

- Maximization of this approximate marginal likelihood then leads to a re-estimated value for $\boldsymbol{\alpha}$, and the process is repeated until convergence.

RVM For Classification

- This can be done using iterative reweighted least squares (IRLS).
- We need the gradient and Hessian of the log posterior distribution

$$\nabla \ln p(\boldsymbol{w}|\boldsymbol{t},\boldsymbol{\alpha}) = \boldsymbol{\Phi} T(\boldsymbol{t} - \boldsymbol{y}) - \boldsymbol{A}\boldsymbol{w}$$
$$\nabla\nabla \ln p(\boldsymbol{w}|\boldsymbol{t},\boldsymbol{\alpha}) = -(\boldsymbol{\Phi}^T \boldsymbol{B}\boldsymbol{\Phi} + \boldsymbol{A})$$

- **B** is an $N \times N$ diagonal matrix with elements $b_n = y_n(1 - y_n)$, the vector $\boldsymbol{y} = (y_1, \ldots, y_N)^T$, and $\boldsymbol{\Phi}$ is the design matrix with elements $\Phi_{ni} = \phi_{i(\boldsymbol{x}_n)}$.
- At convergence of the IRLS algorithm, the negative Hessian represents the inverse covariance matrix for the Gaussian approximation to the posterior distribution.
- The mean and covariance of the Laplace approximation in the form

$$\boldsymbol{w} = \boldsymbol{A}^{-1}\boldsymbol{\Phi}^T(\boldsymbol{t} - \boldsymbol{y})$$
$$\boldsymbol{\Sigma} = (\boldsymbol{\Phi}^T \boldsymbol{B}\boldsymbol{\Phi} + \boldsymbol{A})^{-1}$$

RVM For Classification

- The marginal likelihood can be approximated as:

$$p(\boldsymbol{t}|\boldsymbol{\alpha}) = \int p(\boldsymbol{t}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w} \approx p(\boldsymbol{t}|\boldsymbol{w}^*)p(\boldsymbol{w}^*|\boldsymbol{\alpha})(2\pi)^{M/2}|\boldsymbol{\Sigma}|^{1/2}$$

- If we substitute for $p(\boldsymbol{t}|\boldsymbol{w}^*)$ and $p(\boldsymbol{w}^*|\boldsymbol{\alpha})$ and then set the derivative wrt equal to zero:

$$-\frac{1}{2}(w_i^*)^2 + \frac{1}{2\alpha_i} - \frac{1}{2}\Sigma_{ii} = 0$$

- Defining $\gamma_i = 1 - \alpha_i\Sigma_{ii}$, gives:

$$\alpha_i^{new} = \frac{\gamma_i}{\left(w_i^*\right)^2}$$

- With $\hat{\boldsymbol{t}} = \boldsymbol{\Phi}\boldsymbol{w}^* + \boldsymbol{B}^{-1}(\boldsymbol{t} - \boldsymbol{y})$, we can write

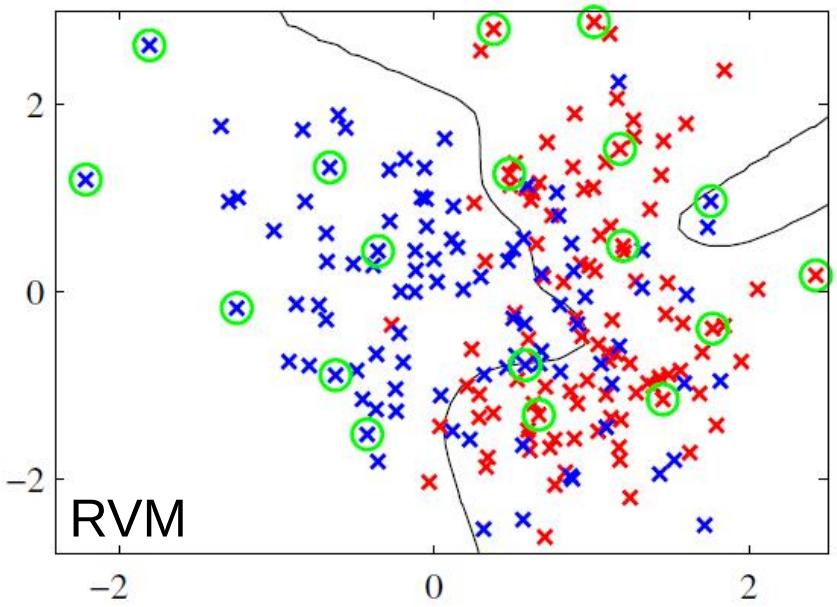
$$lnp(\boldsymbol{t}|\boldsymbol{\alpha}) = -\frac{1}{2}\{Nln(2\pi) + \ln|\boldsymbol{C}| + (\hat{\boldsymbol{t}})^T\boldsymbol{C}^{-1}\hat{\boldsymbol{t}}\},\ \boldsymbol{C} = \boldsymbol{B} + \boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Phi}^T$$

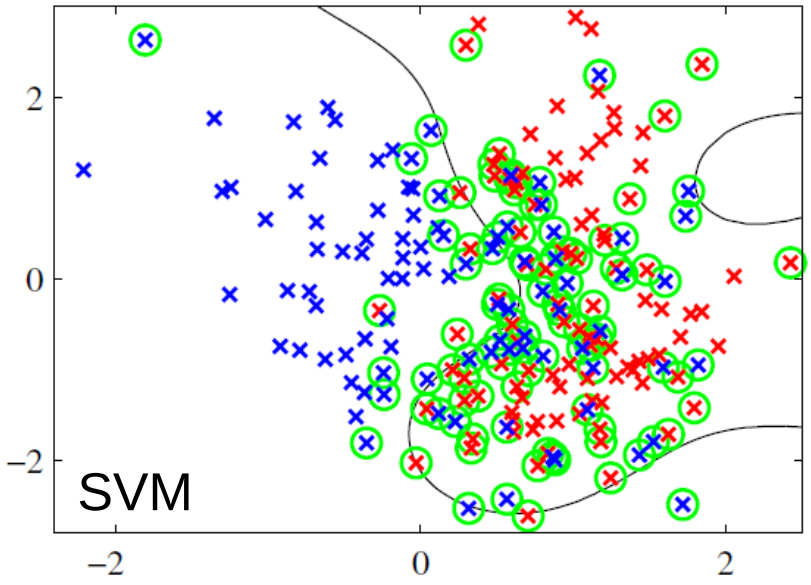
- This is the same form as for regression. Can apply the same sparsity analysis and obtain the same fast learning algorithm in which we optimize a single α_i at each step.

RVM For Classification

- Example of the RVM applied to a synthetic data set, in which the left-hand plot shows the decision boundary and the data points, with the relevance vectors indicated by circles.

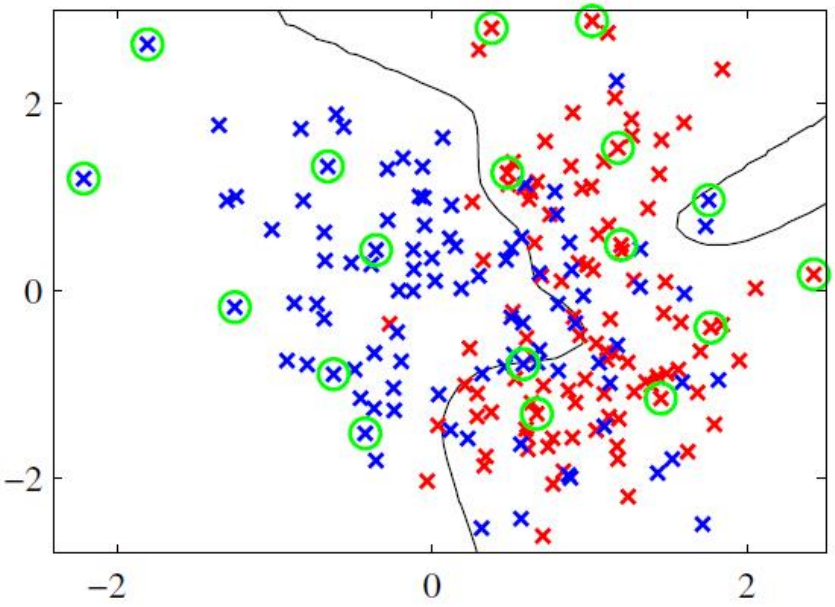
- Comparing with the SVM results it is seen that the RVM gives a much sparser model.

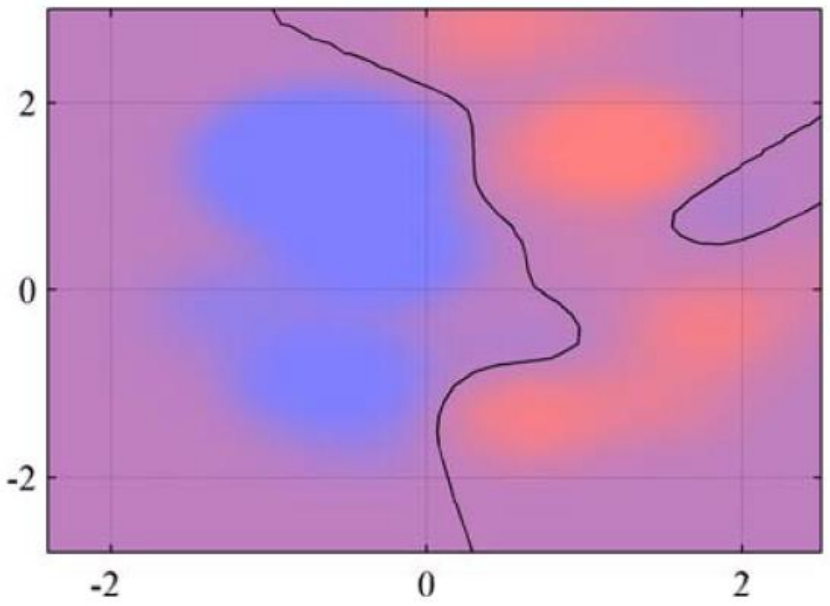




RVM For Classification

- The right-hand plot shows the posterior probability given by the RVM output in which the proportion of red (blue) ink indicates the probability of that point belonging to the red (blue) class.





RVM For Classification

- So far, we have considered the RVM for binary classification problems. For $K > 2$ classes,

$$y_k(\boldsymbol{x}) = \frac{\exp(\alpha_k)}{\sum_j \exp(\alpha_k)}$$

- The log likelihood function is then given by

$$lnp(\boldsymbol{T}|\boldsymbol{w}_1, \dots, \boldsymbol{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

where the target values t_{nk} have a 1-of-K coding for each data point n, and $\boldsymbol{T}$ is a matrix with elements t_{nk}. Again, the Laplace approximation can be used to optimize the hyperparameters in which the model and its Hessian are found using IRLS.

RVM For Classification

- The principal disadvantage is that the Hessian matrix has size $MK \times MK$, where M is the number of active basis functions, which gives an additional factor of K^3 in the computational cost of training compared with the two-class RVM.

- The principal disadvantage of the relevance vector machine is the relatively long training times compared with the SVM. This is offset, however, by the avoidance of cross-validation runs to set the model complexity parameters.

- Furthermore, because it yields sparser models, the computation time on test points, which is usually the more important consideration in practice, is typically much less.

References

- T. Fletcher, Relevance Vector Machines Explained, October 2010.
- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244.
- MacKay, D. (1995b). Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network*.
- Neal, R. (1996). *Bayesian learning for neural networks*. Springer.
- Bishop, C. and M. Tipping (2000). Variational relevance vector machines. In *UAI*.
- Buntine, W. and A. Weigend (1991). Bayesian backpropagation. *Complex Systems 5*, 603–643
- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244.
- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).
- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).
- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*.
- MacKay, D. (1999). Comparision of approximate methods for handling hyperparameters. *Neural Computation 11*(5), 1035–1068.
- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*
- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).

References

- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*.
- Rasmussen, C. E. and J. Qui˜nonero-Candela (2005). Healing the relevance vector machine by augmentation. In L. D. Raedt and S. Wrobel (Eds.), *Proceedings of the 22nd International Conference on Machine Learning*, pp. 689–696.
- T. Fletcher, Relevance Vector Machines Explained, October 2010.
- Faul, A. C. and M. E. Tipping (2002). Analysis of sparse Bayesian learning. In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems*, Volume 14, pp. 383–389. MIT Press.
- Tipping, M. E. and A. Faul (2003). Fast marginal likelihood maximization for sparse Bayesian models. In C. M. Bishop and B. Frey (Eds.), *Proceedings Ninth International Workshop on Artificial Intelligence and Statistics, Key West, Florida*.
- Williams, O., A. Blake, and R. Cipolla (2005). Sparse Bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1292–1304.